



Escola Universitària d'Enginyeria  
Tècnica Industrial de Barcelona  
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

**Memoria**



Barcelona, 13 de Junio de 2012

Director: Antoni Pérez Poch  
Departamento de LSI (723)  
Universitat Politècnica de Catalunya (UPC)



# ÍNDICE MEMORIA

RESUM .....	5
RESUMEN .....	5
ABSTRACT.....	6
AGRADECIMIENTOS.....	7
CAPÍTULO 1: INTRODUCCIÓN.....	9
1.1.    Objetivos: solución propuesta.....	9
1.2.    Metodología.....	10
1.3.    Ámbito y alcance.....	11
CAPÍTULO 2: ANTECEDENTES.....	13
2.1.    Historia de los videojuegos .....	13
2.1.1.    Inicios.....	13
2.1.2.    El auge de los videojuegos.....	16
2.1.3.    La generación de los 8 bits.....	18
2.1.4.    La generación de los 16 bits .....	26
2.1.5.    La generación de los 32 bits .....	29
2.1.6.    La generación de los 64 bits .....	31
2.1.7.    La generación de los 128 bits.....	32
2.1.8.    Consolas de séptima generación (actual) .....	34
2.1.9.    Futuras generaciones.....	37
2.2.    Proyectos de partida.....	38
2.2.1.    Termóstato digital elaborado con microcontrolador .....	39
2.2.2.    Pocket Pong.....	39
2.2.3.    Uzebox .....	40
2.2.4.    Fuzebox .....	40
CAPÍTULO 3: NORMATIVA Y ASPECTOS LEGALES.....	43
3.1.    Normativa referente al <i>hardware</i> .....	43
3.1.1.    Directiva 2002/95/CE (RoHS).....	43
3.1.2.    Compatibilidad electromagnética .....	44
3.1.3.    Directiva 91/157/CEE (Baterías) .....	44
3.1.4.    Directiva 2002/96/CE (WEEE) .....	45
3.2.    Normativa referente al <i>software</i> .....	45
3.2.1.    Copyright.....	45

3.2.2.	<i>Licencias de uso del software utilizado</i> .....	46
CAPÍTULO 4: INGENIERÍA DE CONCEPCIÓN .....		47
4.1.	Solución propuesta .....	47
A continuación se detallará la solución escogida para desarrollar este proyecto. ....		47
4.1.1.	<i>Visualización</i> .....	47
4.1.2.	<i>Microcontrolador</i> .....	48
4.1.3.	<i>Mandos</i> .....	50
4.1.4.	<i>Plataforma de desarrollo</i> .....	51
4.2.	Esquemas iniciales .....	51
4.2.1.	<i>Visualización</i> .....	52
4.2.2.	<i>Controles del videojuego</i> .....	53
4.2.3.	<i>Microcontrolador</i> .....	53
4.3.	Juego a implementar.....	54
CAPÍTULO 5: INGENIERÍA DE DETALLE .....		55
5.1.	Esquemas de detalle del <i>hardware</i> .....	55
5.1.1.	<i>Controles de juego</i> .....	56
5.1.2.	<i>Filtrado</i> .....	56
5.1.3.	<i>Microcontrolador</i> .....	57
5.1.4.	<i>Adecuación de la señal</i> .....	57
5.1.5.	<i>Protecciones</i> .....	58
5.1.6.	<i>Sonido</i> .....	58
5.1.7.	<i>Visualización</i> .....	59
5.1.8.	<i>Otros</i> .....	60
5.2.	Funcionamiento de un <i>LCD</i> gráfico .....	60
5.3.	Selección de componentes y cálculos justificativos.....	64
5.3.1.	<i>Altavoz</i> .....	64
5.3.2.	<i>Resistencia LED</i> .....	65
5.3.3.	<i>Zumbador</i> .....	65
5.3.4.	<i>Condensadores de los pulsadores</i> .....	66
5.4.	Diseño del <i>software</i> .....	67
5.4.1.	<i>Entorno del CCS C</i> .....	67
5.4.2.	<i>Librerías del CCS C</i> .....	69

5.4.3.	<i>Configuración del PIC</i> .....	71
5.4.4.	<i>Diagrama de flujo Deadly Pong</i> .....	72
5.4.5.	<i>Rutinas y Subrutinas de Deadly Pong</i> .....	72
5.4.6.	<i>Imágenes del Deadly Pong</i> .....	75
5.4.7.	<i>Melodías del Deadly Pong</i> .....	77
5.5.	Entorno de grabación del PIC .....	77
CAPÍTULO 6: CRONOGRAMA DE EJECUCIÓN .....		79
6.1.	Programa de ejecución .....	79
6.2.	Estimación de tiempo.....	80
CAPÍTULO 7: PRESUPUESTO .....		81
7.1.	Costes de los materiales .....	81
7.2.	Costes de la ingeniería .....	83
7.3.	Otros costes .....	84
7.4.	Costes totales.....	86
CAPÍTULO 8: POSIBLES MEJORAS Y CONCLUSIONES.....		87
8.1.	Posibles mejoras.....	87
8.1.1.	<i>Mejoras en hardware</i> .....	87
8.1.2.	<i>Mejoras en el software</i> .....	88
8.2.	Conclusión personal.....	88
CAPÍTULO 9: BIBLIOGRAFÍA.....		89
9.1.	Referencias bibliográficas .....	89
9.2.	Bibliografía de consulta .....	90
9.2.1.	<i>Artículos de revistas</i> .....	90
9.2.2.	<i>Libros</i> .....	90
ANEXO A: ESQUEMAS Y PLANOS .....		91



## **RESUM**

Aquest document conté el disseny d'un videojoc propi dels anys 70, utilitzant tecnologia actual. En concret s'ha dut a terme el disseny, tant del hardware com del software, d'un videojoc anomenat Deadly Pong, inspirat en un dels grans èxits que van sorgir en aquella època i que tothom coneix, el Pong.

El videojoc consisteix en controlar una pala per impedir que la pilota passi del nostre terreny, fent-nos perdre. A mesura que es progressa en la partida, es va avançant de nivell, amb el conseqüent augment de dificultat. El videojoc té un total de cinc nivells, i guanyarem si aconseguim passar-los tots.

La novetat d'aquesta videoconsola envers les de fa 40 anys, és que s'ha implementat utilitzant la tecnologia dels microcontroladors, que és de les més utilitzades actualment en el món de l'electrònica. En concret del control s'encarrega un microcontrolador PIC, les ordres s'introdueixen mitjançant un comandament format per polsadors, de la visualització s'encarrega un LCD gràfic de 128x64 punts, i la música es reproduïda amb un petit altaveu.

Aquest disseny és un bon punt de partida per desenvolupar videojocs de més complexitat utilitzant aquesta tecnologia.

## **RESUMEN**

Este documento contiene el diseño de un videojuego propio de los años 70, utilizando tecnología actual. En concreto se ha llevado a cabo el diseño, tanto del hardware como del software, de un videojuego llamado Deadly Pong, inspirado en uno de los grandes éxitos que surgieron en aquella época y que todo el mundo conoce, el Pong.

El videojuego consiste en controlar una pala para impedir que la pelota pase de nuestro terreno, haciéndonos perder. A medida que se progresa en la partida, se va avanzando de nivel, con el consiguiente aumento de dificultad. El videojuego tiene un total de cinco niveles, y ganaremos si conseguimos pasarlos todos.

La novedad de esta videoconsola frente a las de hace 40 años, es que se ha implementado utilizando la tecnología de los microcontroladores, que es de las más utilizadas actualmente en el mundo de la electrónica. En concreto del control se encarga un microcontrolador PIC, las ordenes se introducen mediante un mando formado por pulsadores, de la visualización se encarga un LCD gráfico de 128x64 puntos, y la música es reproducida con un pequeño altavoz.

Este diseño es un buen punto de partida para desarrollar videojuegos de más complejidad utilizando esta tecnología.

## **ABSTRACT**

This document contains the design of a videogame typical of the 70s, using actual technology. Concretely, it has been carried out the design, both hardware and software, of a videogame called Deadly Pong, which is inspired in one of the most successful games of that time and known by everybody, the Pong.

The videogame consists in controlling the paddle to don't allow the ball passes through your field, making us lose. According as we are progressing in the game, we will rise up level, and the difficult will rise up too. The videogame has five levels in total, and we will win whether we can overcome all of them.

The innovation of that videogame against the videogames of 40 years ago, is that it has been implemented using the microcontrollers technology, which is one of the most used technologies in the world of the electronics nowadays. Concretely, the control is handled by a PIC microcontroller, the orders are entered by a pad formed by buttons, a graphical LCD of 128x64 dots is used as a display, and finally, the music is played with a little speaker.

This design is a good start point to develop more complex videogames using that technology.



## **AGRADECIMIENTOS**

Me gustaría agradecer a toda la comunidad universitaria el apoyo prestado durante el desarrollo de este proyecto, tanto de alumnos como de profesores. En especial a todos mis compañeros, que me han animado a seguir adelante y han aportado su granito de arena con útiles consejos y buenos momentos. También merecen un agradecimiento los técnicos del laboratorio de electrónica de la EUETIB, que me han sabido resolver dudas puntuales sobre el diseño y han aportado ayuda en momentos oportunos.

Merecen un especial agradecimiento a parte, mi novia, mis amigos y mi familia, puesto que han vivido conmigo la elaboración de este proyecto y me han proporcionado su total apoyo.

Finalmente, querría dar las gracias a mi tutor de proyecto Antoni Pérez, por aceptarme como proyectista y permitirme presentar este proyecto, así como por toda la orientación, ayuda, e ideas que me ha proporcionado.



# **CAPÍTULO 1: INTRODUCCIÓN**

Este proyecto consiste en el proyecto final de grado (PFG) de los estudios de Grado en Ingeniería Electrónica Industrial y Automática. En él se aplican todos los conocimientos obtenidos mediante la realización de la carrera en el diseño y la construcción de un prototipo de videojuego típico de los años 80 con tecnología moderna.

## **1.1. Objetivos: solución propuesta**

El proyecto a realizar consiste en el estudio y posterior desarrollo y montaje de una videoconsola con las herramientas que se han estudiado durante el grado en ingeniería electrónica industrial y automática cursado por el estudiante.

Se plantea el desafío del diseño y la construcción física del dispositivo, así como el de la programación y síntesis de un videojuego basado en el estilo de los videojuegos de los años 70. En concreto se pretende diseñar un videojuego basado en el Pong publicado por Atari en 1972, creado por Nolan Bushnell.

Respecto al hardware, se pretende realizar la construcción de un prototipo de videoconsola. Dicho prototipo estará controlado por un microcontrolador, su visualización se hará mediante un visualizador gráfico de cristal líquido (GLCD), y además podrá reproducir melodías durante la partida con una etapa de audio.

Así pues lo que se pretende es ganar experiencia tanto en el diseño de hardware como de software y familiarizarse con el uso de microcontroladores y displays LCD.

## 1.2. Metodología

La metodología a seguir durante la elaboración del proyecto será la siguiente:

- Búsqueda de información sobre el estado del arte, así como de proyectos similares y/o relacionados.
- Realizar un estudio previo de las características del diseño (Analizar objetivos, Identificación de herramientas necesarias, viabilidad económica, etc.).
- Selección de componentes adecuados a las características del diseño y elaboración de un esquema electrónico inicial.
- Obtención y estudio de las principales herramientas de software necesarias para la programación y simulación del videojuego.
- Diseño y compilación del videojuego en un lenguaje de programación en alto nivel, concretamente en C.
- Simulación del videojuego en el esquema electrónico mediante software especializado. Corrección de errores y elaboración del esquema electrónico definitivo.
- Adquisición de los componentes, comprobación y montaje (Diseñar la placa de circuito impreso, comprobar el funcionamiento de los componentes, realizar el montaje, comprobar conexiones, etc.).
- Carga del videojuego en la memoria caché del microcontrolador y comprobación de su funcionamiento.
- Detección y corrección de errores.
- Estudio e implementación de posibles mejoras en el prototipo.

Transversalmente se elabora una memoria de proyecto que recoja todos los detalles del diseño, y que posteriormente pueda ser consultada por terceras personas con el objetivo de realizar proyectos similares o incluso de repetir el mismo.

### 1.3. Ámbito y alcance

El objetivo final de este proyecto es, como se ha dicho anteriormente, la construcción de un prototipo de videoconsola, para consolidar los conocimientos obtenidos dentro del campo de la electrónica y de la informática, así pues su ámbito es puramente didáctico.

Dicho proyecto contempla todo el diseño y construcción del prototipo necesario para la ejecución de un videojuego, así como su programación.

En cuanto al alcance, el videojuego final desarrollado no pretende ser objeto de distribución de ningún tipo, siendo un proyecto de alcance personal. Sin embargo, se permite su total uso o reproducción para ser compartido como una herramienta de autoaprendizaje que permita al usuario adquirir o aumentar sus conocimientos sobre programación y electrónica. Para este propósito se facilitan todos los esquemas y planos utilizados así como el código de programa realizado, para permitir su libre recreación y/o modificación.



# CAPÍTULO 2:

## ANTECEDENTES

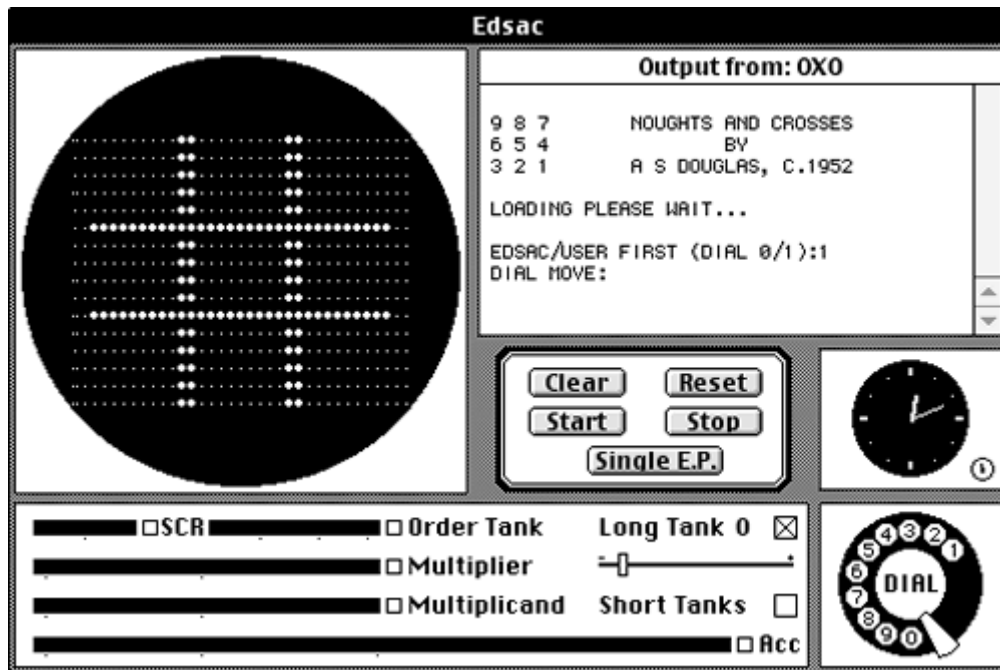
Se procede a realizar un repaso a la historia de los videojuegos, desde sus orígenes, hasta la actualidad, haciendo especial hincapié en aquellos que han servido de inspiración para el desarrollo del videojuego realizado. También se observaran proyectos similares que pueden servir de punto de partida para el diseño del *hardware*.

### 2.1. Historia de los videojuegos

La evolución de los videojuegos ha ido de la mano del desarrollo de las nuevas tecnologías electrónicas, dando lugar incluso a la creación de una industria dedicada íntegramente a los videojuegos que en 2011 facturó 980 millones de euros solo en territorio Español.

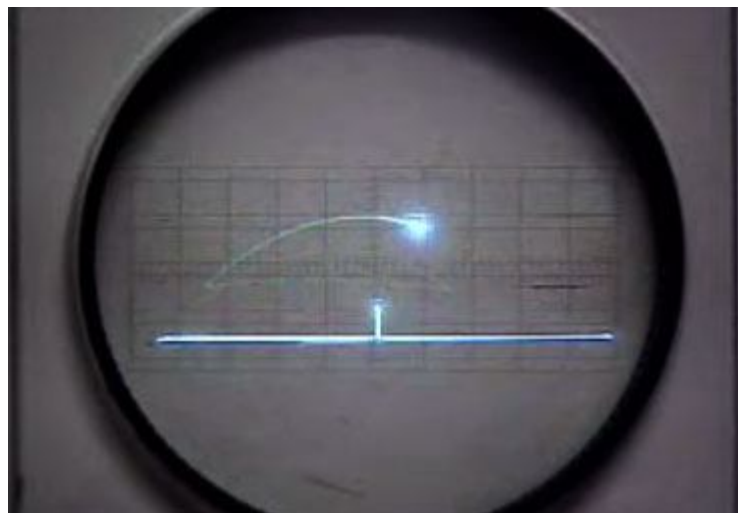
#### 2.1.1. *Inicios*

El primer videojuego electrónico del que se tiene constancia fue desarrollado en 1952 por Alexander Sandy Douglas. Llamado *Naught and Crosses*, aunque más comúnmente conocido como OXO, era una versión computerizada del tres en ralla que solamente se podía ejecutar en uno de los ordenadores de la época, el EDSAC, y permitía medirse al jugador contra la máquina. Alexander programó el juego como parte de una investigación para la Universidad de Cambridge sobre la interacción de las máquinas con el ser humano. El juego no alcanzó mucha popularidad en su entorno, ya que el ordenador EDSAC era único. Pese a que anteriormente existían videojuegos en modo texto, el OXO es considerado el primer videojuego con gráficos de la historia.



**Figura 1.** OXO (fuente: <http://www.pixfans.com/oxo-el-primer-videojuego-de-la-historia/>).

En 1958 William Higginbotham creó el *Tennis for Two*: un simulador de tenis de mesa para entretenimiento de los visitantes de la exposición *Brookhaven National Laboratory*. El videojuego se valía de un programa de cálculo de trayectorias y de un osciloscopio.

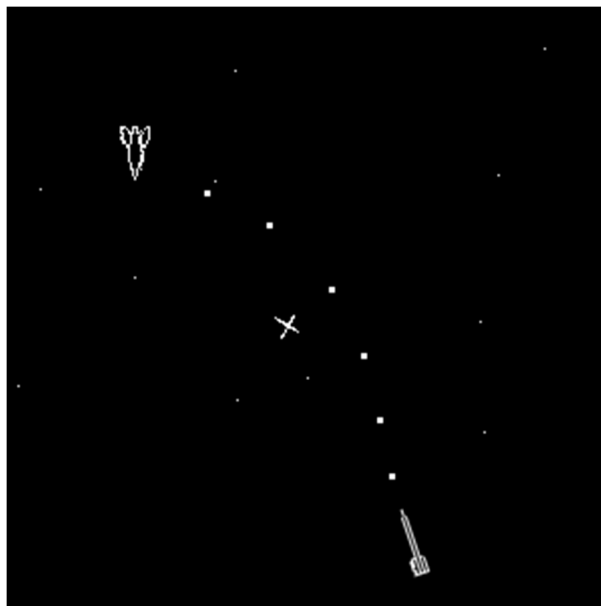


**Figura 2.** *Tennis for Two* (fuente: [http://gamecodelegend.wikia.com/wiki/Tennis\\_For\\_Two](http://gamecodelegend.wikia.com/wiki/Tennis_For_Two)).

En 1962 Steve Russell y dos compañeros suyos, todos estudiantes del MIT (Instituto Tecnológico de Massachusetts), dedicaron seis meses en crear un juego por ordenador usando gráficos vectoriales: el *Spacewar*. En este juego, dos jugadores controlaban la dirección y la velocidad de dos naves espaciales que luchaban entre ellas. El videojuego funcionaba sobre una



máquina PDP-1 y fue el primero en tener cierto éxito pese a que fue poco conocido fuera del ámbito universitario.

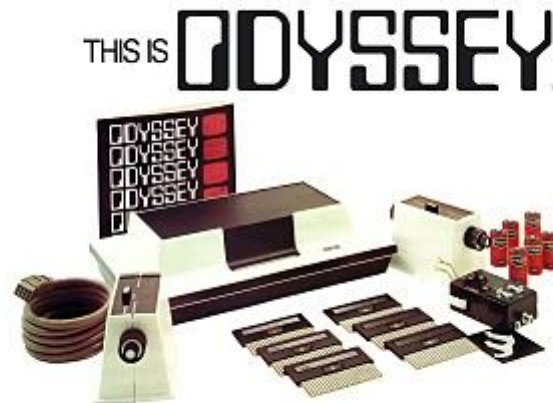


**Figura 3.** *Spacewar* (fuente: <http://es.wikipedia.org/wiki/Spacewar!>).

Según muchos, el origen de los videojuegos en su concepción actual y el inicio del entretenimiento electrónico fue en 1966. Ralph Baer cambia su idea original de "televisiones con juegos" y lo transforma en "cajas de juegos" (conocidos posteriormente como videoconsolas o simplemente consolas).

Baer y un equipo de desarrolladores crearon un juego llamado *Fox and Hounds*, en el cual dos personas controlaban dos puntos que simulaban a estos animales, y donde el uno debía perseguir al otro. El *Fox and Hounds* es considerado el primer videojuego en toda la extensión de la palabra.

Para poder jugar a *Fox and Hounds* y otros juegos, Baer también crearía el "Brown Box", el cual fue lanzado al mercado como el *Magnavox Odyssey* en 1972, dando así nacimiento a la primera consola casera de todos los tiempos. Esta consola no poseía memoria para almacenar la puntuación por lo que los jugadores debían memorizar su puntuación. Debido a la complejidad de su uso, no fue muy exitosa.



**Figura 4.** Magnavox Odyssey (fuente: <http://gamersup.blogspot.com.es/2011/02/magnavox-odyssey.html>).

### 2.1.2. El auge de los videojuegos

Mientras Ralph Baer trabajaba con el *Magnavox Odyssey*, en 1972 el programador Nolan Bushnell funda la compañía *Atari* y comercializa la máquina recreativa *Pong*, el cual era una versión moderna del antiguo *Tennis for Two* creado en osciloscopio por William Higginbotham. Dicha máquina recreativa contaba con circuitos integrados lógicos internos y una pantalla de televisión.

Esta máquina tuvo una gran acogida y le sirvió tecnológica y económicamente a Bushnell para lanzar posteriormente varias consolas domésticas.

Este juego es considerado como la piedra angular del videojuego como industria. Durante los años siguientes se implementaron numerosos avances técnicos en los videojuegos entre los que destacan los microprocesadores y los chips de memoria, y aparecieron, en salones recreativos, videojuegos como *Space Invaders* (Taito) o *Asteroids* (Atari).



**Figura 5.** Pong (fuente: <http://es.wikipedia.org/wiki/Pong>).

En 1975 se lanzaría la versión doméstica del *Pong*. Esta consola tuvo un gran impacto cultural y mucha popularidad. La máquina solo permitía jugar a *Pong* y disponía de 2 controladores incorporados.

Durante 1975 y 1977 *Atari* lanzó varias versiones del *Pong* con diferentes innovaciones como color o puntuación digital en la pantalla



**Figura 6.** Pong doméstico (fuente: <http://dekagamers.blogspot.com.es/2012/04/retro-consolas-primera-generacion-vol.html>).

### 2.1.3. La generación de los 8 bits

Finalmente en 1977 nace la *Atari VCS (Video Computer System)* que después se la conocería como *Atari 2600*. Esta sería la primera consola de 8 bits y la que finalmente revolucionaría el mercado de los videojuegos.

Utilizaba cartuchos intercambiables para todo tipo de juegos, con una galería de más de 2000 títulos, dando aquí nacimiento a muchos de los géneros que se mantienen vigentes hasta la actualidad.

Los géneros de deportes, plataformas, aventura, *shooter*, las adaptaciones de películas a videojuegos, etc. vieron su nacimiento y desarrollo en esta consola.

Así surgieron juegos como: *Pacman*, *Defender*, *Adventure*, *Donkey Kong*, *E.T.*, *Atari Soccer*, *Chess*, *Enduro*, *Pitfall*, etc.

La *Atari 2600* fue la consola más popular hasta el surgimiento del *Nintendo Entertainment System (NES)* y estuvo en el mercado durante 14 años convirtiéndose así en la consola con mayor tiempo de vida de toda la historia (se la comercializo hasta 1990).



**Figura 7.** Atari 2600 (fuente: [http://en.wikipedia.org/wiki/Atari\\_2600](http://en.wikipedia.org/wiki/Atari_2600)).

En 1978 surge el *Odyssey2*, el cual era el sucesor del *Odyssey* original. Esta consola poseía una tecnología similar a la de la *Atari 2600* además disponía de un teclado incorporado (algo muy innovador en las consolas de esta generación).

Tuvo bastante acogida y fue una de las más grandes competencias para *Atari*, aunque nunca logro superarla en popularidad.



**Figura 8.** Odyssey2 (fuente: <http://xor2086.tumblr.com/post/675224529/pepeto-papupapa-y-el-odyssey2>).

La *Intellivision*, distribuida por *Mattel Electronics*, aparece en 1980 con una tecnología superior a la de la *Atari 2600* y con la promesa de convertirse en un ordenador doméstico.

Esta consola se vendió durante 10 años por correo, sin embargo la deficiencia en esta forma de venta y el poco la poca publicidad hicieron que no alcanzara demasiada popular.



**Figura 9.** Intellivision (fuente: <http://old-computers.com/museum/computer.asp?st=2&c=1205>).

En 1980 surge el *Commodore VIC-20*, el cual en realidad era un ordenador doméstico adaptado a consola. No alcanzo mucho éxito pero tuvo algunos juegos populares.



**Figura 10.** *Commodore VIC-20* (fuente: [http://es.wikipedia.org/wiki/Commodore\\_VIC-20](http://es.wikipedia.org/wiki/Commodore_VIC-20)).

Durante 1982 apareció el *Vectrex*. Esta fue la primera y única consola en la historia que utilizó vectores gráficos para los videojuegos. Nunca logró ser muy popular aunque su valor histórico es bastante importante.



**Figura 11.** *Vectrex* (fuente: <http://es.wikipedia.org/wiki/Vectrex>).

La compañía *Emerson* lanzó en 1982 la consola *Arcadia 2001*. Pese a que fue bastante popular e hizo una fuerte competencia a la *Atari 2600*, no estuvo mucho tiempo en el mercado.





**Figura 12.** Arcadia 2001 (fuente: <http://curiosidadesybasura.blogspot.com.es/2011/07/consolas-que-dadas-en-el-olvido-arcadia.html>).

En 1982 nace la *ColecoVision*. Esta consola tenía una tecnología gráfica y una calidad de sonido superior a todas las consolas de ese tiempo y se la comercializo junto al popular juego *Donkey Kong*. La *Coleco Vision* fue la máxima competencia para la *Atari 2600*.



**Figura 13.** ColecoVision (fuente: <http://www.vootar.com/a/ColecoVision>).

Durante 1982, la compañía *Atari* lanza la *Atari 5200* la cual era la sucesora de la famosa *Atari 2600*. Esta consola tenía varios fallos entre las que se incluían malos controles y que no era compatible con los cartuchos de la *Atari 2600*, por lo que no tuvo buena acogida y no pudo remplazar a su predecesora.



**Figura 14.** Atari 5200 (fuente: [http://en.wikipedia.org/wiki/Atari\\_5200](http://en.wikipedia.org/wiki/Atari_5200)).

En 1983 se lanzó al mercado el MSX, el cual fue diseñado como un ordenador doméstico, debido a estándares mundiales. Su uso como consola fue popular en Europa y Asia, en especial Japón.



**Figura 15.** MSX (fuente: <http://en.wikipedia.org/wiki/MSX>).

Durante los años de éxito de *Atari*, la compañía lanzó varios ordenas al mercado, entre los que están los *Atari 400xl*, *Atari 600xl*, y *Atari 1200xl*, pero en 1983, se lanza el *Atari 800xl* en un intento de remplazar todos los ordenadores anteriores. Todos ellos se utilizaron también como videoconsolas. Esta consola era similar a la *Atari 1200xl* pero tenía opciones diferentes. A pesar de que *Atari* sacó al mercado estas nuevas consolas y nueva tecnología, ninguna consiguió ser tan popular como la *Atari 2600*.





**Figura 16.** Atari 800xl (fuente: [http://es.wikipedia.org/wiki/Atari\\_800XL](http://es.wikipedia.org/wiki/Atari_800XL)).

Entre 1983 y 1985 hubo una crisis en el sector que afectó principalmente a los Estados Unidos y a Canadá. Este hecho propició que se desarrollaran potentes videoconsolas en otros países.

En Japón se desarrolló la *Famicom*, conocida en occidente como la *NES* (*Nintendo Entertainment System*), y que salió al mercado en el año 1983, mientras Europa se decantaba por los microordenadores como el *Commodore 64* o el *Spectrum*. La *NES* fue la encargada de destronar a la *Atari 2600* como la videoconsola más popular, y está considerada como la videoconsola más exitosa de su época.



**Figura 17.** NES (fuente: <http://es.wikipedia.org/wiki/NES>).

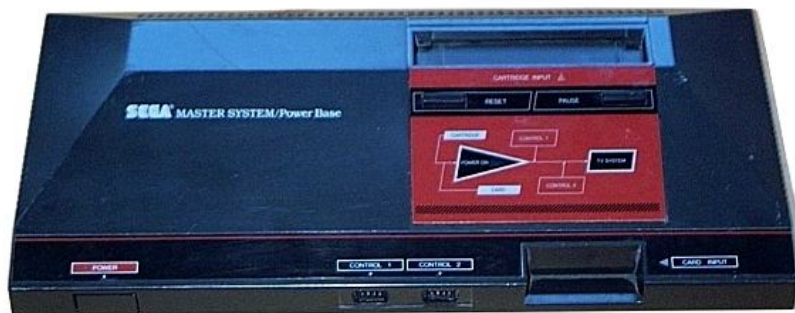
El *Sinclair ZX Spectrum* fue un ordenador de 8 bits basado en el microprocesador *Zilog Z80A*, fabricado por la compañía británica *Sinclair Research* y lanzado al mercado el 23 de abril de 1982. En Europa, el *Sinclair ZX Spectrum* fue uno de los microordenadores domésticos más populares de los años 1980.

Su optimizado y compacto diseño hizo las delicias de miles de aficionados a la informática y los videojuegos.



**Figura 18.** Sinclair ZX Spectrum de 48K (fuente: [http://es.wikipedia.org/wiki/Sinclair\\_ZX\\_Spectrum](http://es.wikipedia.org/wiki/Sinclair_ZX_Spectrum)).

A lo largo de la década de los 80 aparecieron nuevos sistemas domésticos como la *Master System* (Sega), el *Amiga* (Commodore) y la *Atari 7800* con juegos considerados hoy día como clásicos como es el caso del *Tetris*.



**Figura 19.** Sega Master System (fuente: [http://es.wikipedia.org/wiki/Sega\\_Master\\_System](http://es.wikipedia.org/wiki/Sega_Master_System)).



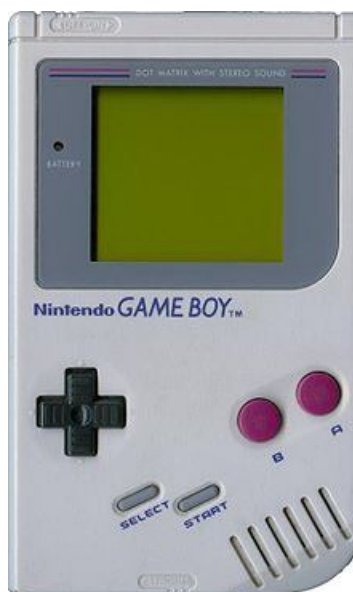
**Figura 20.** Atari 7800 (fuente: [http://es.wikipedia.org/wiki/Atari\\_7800](http://es.wikipedia.org/wiki/Atari_7800)).

El año 1985 apareció el videojuego *Super Mario Bros*, que supuso un punto de inflexión en el desarrollo de los juegos electrónicos, ya que la mayoría de juegos anteriores solamente contenían unas pocas pantallas que se repetían en un bucle, y el objetivo era simplemente conseguir una puntuación elevada. El juego de *Nintendo* supuso un aumento de creatividad. Por una vez se tenía un objetivo y un final concreto en un videojuego. Los años posteriores las compañías de la competencia emularon este tipo de juego.



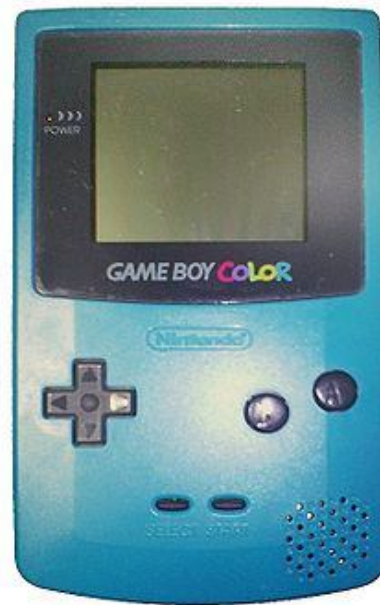
**Figura 21.** *Super Mario Bros* (fuente: [http://www.wikiblogmix.com/como\\_nacio\\_super\\_mario\\_bros](http://www.wikiblogmix.com/como_nacio_super_mario_bros)).

Una rama de los videojuegos que creció mucho, era la de los dispositivos portátiles. Pero la evolución definitiva de las consolas portátiles llegó en 1989 de la mano de *Nintendo* con el lanzamiento de la *Game Boy*.



**Figura 22.** *Nintendo Game Boy* (fuente: [http://es.wikipedia.org/wiki/Game\\_Boy](http://es.wikipedia.org/wiki/Game_Boy)).

El éxito de la *Game Boy* no tuvo precedentes y *Nintendo* supo aprovecharlo sacando nuevas versiones mejoradas de la misma que también cosecharon grandes éxitos. Entre ellas destaca el lanzamiento de la *Game Boy Color* en 1998, que dispone de una pantalla *LCD Sharp* en color, de 160x144 píxeles, capaz de mostrar 32768 colores.



**Figura 23.** Nintendo Game Boy Color (fuente: [http://es.wikipedia.org/wiki/Game\\_Boy\\_Color](http://es.wikipedia.org/wiki/Game_Boy_Color)).

#### 2.1.4. La generación de los 16 bits

A finales de los 80 comenzaron a aparecer consolas de 16 bits como la *Mega Drive* de *Sega* (1988) y los micróordenadores fueron lentamente sustituidos por *PC's* basados en arquitecturas de *IBM*.



**Figura 24.** Sega Mega Drive (fuente: [http://es.wikipedia.org/wiki/Sega\\_Mega\\_Drive](http://es.wikipedia.org/wiki/Sega_Mega_Drive)).

Con la llegada de los 16 bits, las videoconsolas dieron un salto técnico importante sobretodo gracias a la competición llevada por la *Sega Mega*

Drive, la Super NES de Nintendo, la PC Engine de NEC, y la CPS Changer de Capcom.



**Figura 25.** Super NES (o Super Nintendo) (fuente: [http://es.wikipedia.org/wiki/Super\\_NES](http://es.wikipedia.org/wiki/Super_NES)).

Junto con ellas también apareció la Neo Geo (SNK) en 1990, una consola que igualaba en prestaciones técnicas a una máquina recreativa pero demasiado cara para poder llegar de forma masiva a los consumidores.



**Figura 26.** NEO-GEO (fuente: <http://www.start-game.com/wiki/Neo-Geo>).

Esta generación provocó un aumento importante en la cantidad de jugadores i la introducción de nuevas tecnologías como el CD-ROM, principalmente gracias a las nuevas capacidades técnicas.

Mientras tanto, diversas compañías habían comenzado a trabajar en videojuegos con entornos tridimensionales, sobre todo en el campo de los ordenadores, y se obtuvieron diferentes resultados, como por ejemplo, las



"2D y media" de *Doom*, las 3D completas de *4D Boxing* y las 3D sobre entornos prerenderizados como en *Alone in the Dark*.



**Figura 27.** *Doom* (fuente: <http://games1ink.blogspot.com.es/2012/05/doom-1-para-pc-1-link.html>).



**Figura 28.** *Alone in the Dark* (fuente: <http://www.shdownloads.com.ar/2012/03/alone-in-dark-1992-en-hd.html>).

Surgió el primer juego poligonal en consola, *Virtual Racing*, para *Sega Mega Drive*, juego que tuvo gran éxito y que marcó un antes y un después en los juegos de 3D.



**Figura 29.** Virtual Racing (fuente: <http://www.museodelvideojuego.com/virtual-racing-1992/>).

#### 2.1.5. La generación de los 32 bits

Rápidamente los videojuegos en 3D ocuparon el lugar predominante en el mercado, esto tuvo que ver con la salida al mercado de las dos principales consolas de 32 bits, la *Sony PlayStation* (1994) y la *Sega Saturn* (1994), siendo la primera la que cosecharía mayores éxitos dada su mayor popularidad y variedad de juegos. Paralelamente, en cuanto a videojuegos en ordenador, cabe mencionar la creación de las aceleradoras 3D que supusieron un gran salto cualitativo la potencia gráfica de los ordenadores.



**Figura 30.** Sony PlayStation (fuente: <http://es.wikipedia.org/wiki/PlayStation>).



**Figura 31.** *Sega Saturn* (fuente: [http://es.wikipedia.org/wiki/Sega\\_Saturn](http://es.wikipedia.org/wiki/Sega_Saturn)).

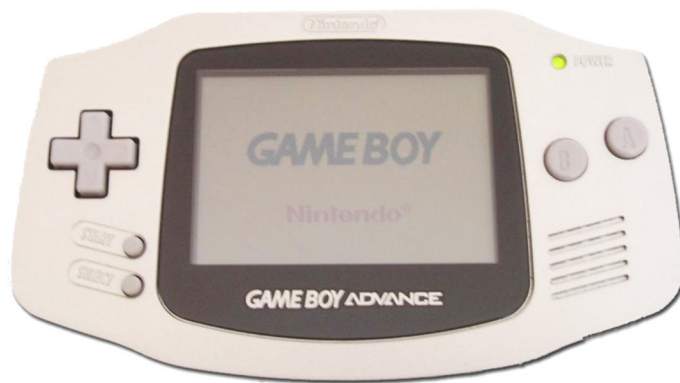
Hasta finales de la década de los 90, la consola más popular fue la PlayStation con juegos como *Final Fantasy VII* (Square), *Resident Evil* (Capcom), *Gran Turismo* (Polyphony Digital) y *Metal Gear Solid* (Konami).



**Figura 32.** *Final Fantasy VII* (fuente: <http://duracion-aproximada.blogspot.com.es/2012/05/analisis-videojuegil-final-fantasy-vii.html>).

En el año 2000, *Nintendo* da un salto en la tecnología de las consolas portátiles sacando la *Game Boy Advance*, la sucesora de la *Game Boy Color*, cosechando un gran éxito en el sector.





**Figura 33.** Game Boy Advance (fuente: [http://es.wikipedia.org/wiki/Game\\_Boy\\_Advance](http://es.wikipedia.org/wiki/Game_Boy_Advance)).

#### 2.1.6. La generación de los 64 bits

Nintendo presentó en 1996 su nueva consola *Nintendo 64*, una máquina de 64 bits que, además de contar con títulos tan conocidos como el *Super Mario 64*, tenía el apoyo de importantes compañías desarrolladoras. La nueva consola de Nintendo, disponía de una tecnología muy por encima de la de sus competidoras (*Sega Saturn* y *Sony PlayStation*), sin embargo sus juegos se comercializaban en cartuchos (un medio mucho más caro que el *CD-ROM*) y en parte por ello la consola no llegó a cuajar entre el público teniendo moderado éxito, pero muy por debajo de la *Sony PlayStation*.



**Figura 34.** Nintendo 64 (fuente: [http://en.wikipedia.org/wiki/Nintendo\\_64](http://en.wikipedia.org/wiki/Nintendo_64)).

### 2.1.7. La generación de los 128 bits

En 1998 *Sega* fue la primera en lanzar una consola de 128 bits, fue la *Sega Dreamcast*. El sistema fue el primero en incorporar un módem para poder jugar *online*. Lamentablemente no pudo competir con sus rivales y *Sega* decidió interrumpir su producción y en adelante dedicarse exclusivamente a producir *software*.



**Figura 35.** *Sega Dreamcast* (fuente: <http://en.wikipedia.org/wiki/Dreamcast>).

En el año 2000 *Sony* lanzó su nueva videoconsola *Sony PlayStation 2* que se convirtió en la videoconsola más vendida de todos los tiempos.



**Figura 36.** *Sony PlayStation 2* (fuente: [http://en.wikipedia.org/wiki/PlayStation\\_2](http://en.wikipedia.org/wiki/PlayStation_2)).

La *PlayStation 2* incorpora una unidad *DVD-ROM* y permite conexión mediante USB, así como formar parte de una red de área local (*LAN*). Fue a su vez la primera videoconsola en incorporar un disco duro.

En 2001 *Nintendo* lanzó su videoconsola para competir con la *PlayStation 2*, la *Nintendo GameCube*. Como novedad respecto a sus predecesoras, se dejó de utilizar cartuchos para los juegos y se utilizaron los llamados *Nintendo Optical Disk*, un formato óptico propio. Pese a ser una videoconsola potente, fracasó en ventas e hizo a *Nintendo* replantearse su estrategia comercial.



**Figura 37.** *Nintendo GameCube* (fuente: [http://en.wikipedia.org/wiki/Nintendo\\_Gamecube](http://en.wikipedia.org/wiki/Nintendo_Gamecube)).

En 2001 *Microsoft* también irrumpe con fuerza en el mundo de las videoconsolas con el lanzamiento de su *Xbox*. La consola incluía disco duro de 8 Gb, *DVD-ROM*, puerto *Ethernet* y procesador dedicado *Intel Pentium III*. Pese a ser nueva en el mercado, irrumpió con fuerza al contar con grandes títulos como *Halo*, sin embargo no logró desbancar en ventas a la *PlayStation 2*, pero sí a la *Nintendo GameCube*.



**Figura 38.** *Xbox* (fuente: <http://en.wikipedia.org/wiki/Xbox>).

### 2.1.8. Consolas de séptima generación (actual)

Llegados a los 128 bits, los desarrolladores dejaron de competir en el número de bits de sus dispositivos, puesto que con 128 bits se ofrecía el punto óptimo entre rendimiento y coste. Así que las generaciones actuales de consolas de sobremesa continúan utilizando configuraciones de 128 bits.

En 2005 *Microsoft* sacó al mercador su nueva videoconsola llamada *Xbox 360*. Sus principales características son su nueva unidad central de procesamiento basado en un *IBM PowerPC* y su nueva unidad de procesamiento gráfico más potente. La *Xbox 360* ofrece juegos renderizados de forma nativa con resoluciones de alta definición. Posteriores versiones incluyen *Wi-Fi* y una arquitectura completamente mejorada. Tuvo una buena acogida en el mercado tanto americano como europeo.



**Figura 39.** *Xbox 360* (fuente: [http://en.wikipedia.org/wiki/Xbox\\_360](http://en.wikipedia.org/wiki/Xbox_360)).

En 2006 salió al mercado la nueva consola de *Sony*, la *PlayStation 3*. La principal característica de este nuevo dispositivo es que ofrece, además de juegos de alta definición, la reproducción de contenido multimedia en alta definición a través de su reproductor de *Blu-ray Disc*. También ofrece un servicio unificado de videojuegos en línea, llamado *PlayStation Network* y conectividad mediante *HDMI*. Al contrario de lo que pasó con sus sucesoras, la *PlayStation 3* no cosechó el gran número de ventas que se esperaba, debido en parte a la competencia y en parte a su alto precio de salida al mercado (alrededor de 600 €). En 2009 salió a la venta su modelo compacto llamado *PlayStation 3 Slim* que incluye un disco duro de mayor capacidad y un menor tamaño.



**Figura 39.** PlayStation 3 y PlayStation 3 Slim (fuente: [http://en.wikipedia.org/wiki/PlayStation\\_3](http://en.wikipedia.org/wiki/PlayStation_3)).

En 2006 *Nintendo* también sacó su nuevo modelo de videoconsola, llamado *Wii*. Su principal novedad es la integración de controladores con sensores de movimiento junto con palancas de mando, revolucionando el mundo de los videojuegos haciendo que el jugador sea parte del mando. Este modelo ha sido imitado posteriormente por sus competidoras. Las cifras de venta de *Nintendo Wii* han sido todo un éxito, devolviendo a la firma a la cima del mundo de los videojuegos.



**Figura 40.** Nintendo Wii (fuente: <http://en.wikipedia.org/wiki/Wii>).

En cuanto a dispositivos portátiles, *Nintendo* lanzó en 2004 su nueva consola portátil, llamada *Nintendo DS (Dual Screen)*. Su principal novedad es que tiene dos pantallas (una de ellas táctil), y permite al jugador interactuar con el juego mediante la pantalla táctil y un micrófono. Ha sido la segunda consola más vendida de todos los tiempos por detrás de la *PlayStation 2*. Posteriormente se han sacado versiones mejoradas, que por ejemplo incluyen cámara de fotos.



**Figura 41.** *Nintendo DS* (fuente: [http://en.wikipedia.org/wiki/Nintendo\\_DS](http://en.wikipedia.org/wiki/Nintendo_DS)).

Viendo en el mercado de las consolas portátiles una oportunidad de negocio, *Sony* lanzó en 2004 una videoconsola llamada *PlayStation Portable (PSP)*. Siendo sus características principales poder reproducir gran cantidad de contenido multimedia, navegar por internet y tener una gran potencia gráfica. Ha sido bien acogida en el mercado compitiendo directamente con *Nintendo DS*.



**Figura 42.** *PSP* (fuente: [http://en.wikipedia.org/wiki/PlayStation\\_Portable](http://en.wikipedia.org/wiki/PlayStation_Portable)).



### 2.1.9. Futuras generaciones

Tanto *Sony* como *Microsoft* han anunciado que habrá una próxima generación de sus videoconsolas en el futuro, *PlayStation 4* y *Xbox 720*, sin embargo solamente ha sido *Nintendo* la que ha anunciado oficialmente una fecha de salida para su nuevo dispositivo. Concretamente *Nintendo* ha anunciado el lanzamiento de su nueva videoconsola *Nintendo Wii U* para las navidades de 2012. Sin embargo aún no se conocen las características definitivas del producto.



**Figura 43.** Prototipo de *Nintendo Wii U* (fuente: <http://www.xataka.com/consolas-y-videojuegos/wii-u-primer-contacto>).

Lo que se ha podido observar a partir del prototipo mostrado es que el sistema dispone de un mando con pantalla incluida, que se comunica vía *Wi-Fi* con la videoconsola. Se sabe que permite reproducir juegos en *HD*, y que incorpora un incremento notable en su potencia visual.

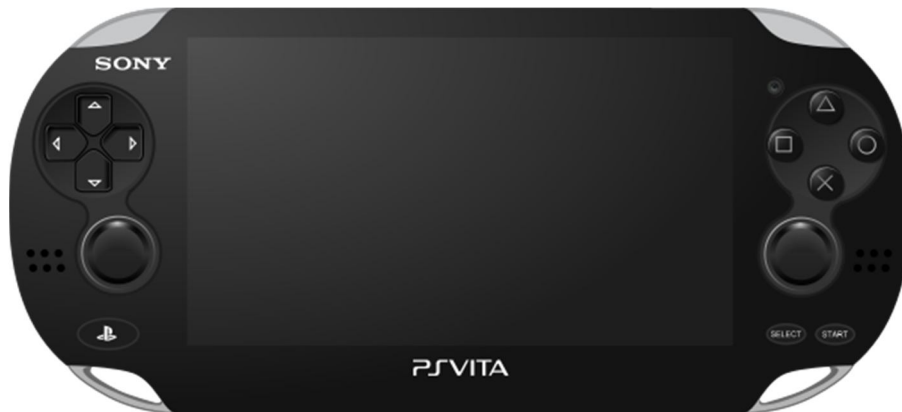
Por otra parte, la siguiente generación de dispositivos portátiles ha salido recientemente al mercado.

En el segundo trimestre de 2011 *Nintendo* puso a la venta su nueva consola portátil, llamada *Nintendo 3DS*. Como su nombre indica es parecida a la anterior versión *Nintendo DS* salvo la novedad de que permite visualizar el videojuego en 3D sin gafas especiales. Pese a la novedad del producto, no ha cosechado el resultado en ventas previsto por *Nintendo* debido principalmente a algunos problemas con este tipo de visualización.



**Figura 44.** Nintendo 3DS (fuente: [http://es.wikipedia.org/wiki/Nintendo\\_3DS](http://es.wikipedia.org/wiki/Nintendo_3DS)).

A finales de 2011 Sony sacó su nueva videoconsola portátil, *PlayStation Vita*. Su principal novedad es que incorpora una pantalla táctil, y es capaz de reproducir cualquier contenido multimedia en HD. Sus desarrolladores afirman que es tan potente como la *PlayStation 3*.



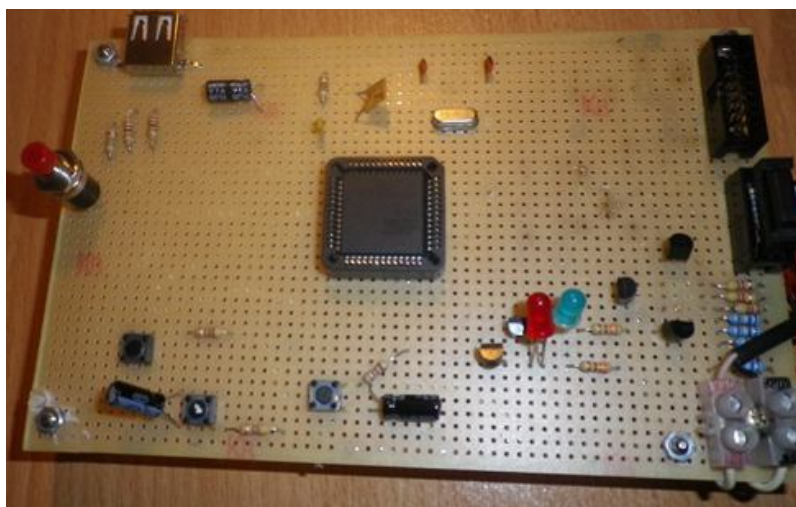
**Figura 45.** PlayStation Vita (fuente: [http://es.wikipedia.org/wiki/PlayStation\\_Vita](http://es.wikipedia.org/wiki/PlayStation_Vita)).

## 2.2. Proyectos de partida

Seguidamente se procede a declarar los proyectos que han sido utilizados como referentes a seguir sobretodo en cuanto a diseño de *hardware*, proporcionando así un modelo de partida.



### 2.2.1. Termóstato digital elaborado con microcontrolador



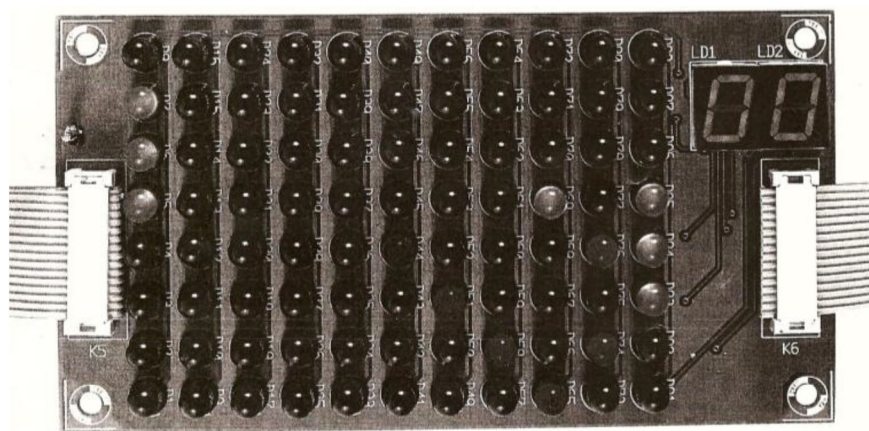
**Figura 46.** Termóstato digital elaborado con microcontrolador (fuente: propia).

Este proyecto fue realizado como parte de una actividad No Presencial propuesta en la asignatura *Informática Industrial (II-EIA)* y fue realizado por el alumno en el cuatrimestre de primavera de 2011.

En él se utiliza un microcontrolador *Atmel 89C5131* para realizar un control de temperatura en una vivienda. El lenguaje de programación utilizado fue el C.

### 2.2.2. Pocket Pong

La implementación de un videojuego con un microcontrolador está inspirada en el montaje realizado en la revista *Elektor*, en el artículo *Pocket Pong*, en el cual se implementa una versión del videojuego *Pong* de *Atari* lanzado en 1972. El montaje se realiza con un *PIC18F452* de *microchip*, y su visualización se basa en una matriz de LEDs.



**Figura 47.** Pocket Pong desarrollado por Elektor (fuente: revista Elektor, número 291).

### 2.2.3. Uzebox

Uzebox es una videoconsola libre de diseño *retro*. Se basa en el microcontrolador AVR de 8 bits de *Atmel*. La particularidad del sistema es que utiliza un *software* controlado por interrupciones y sus gráficos no son representados como ubicaciones en la memoria *RAM*. Las funciones tales como sincronizar la generación del video, renderizados y la mezcla de audio se realizan en tiempo real por una tarea de fondo para que los juegos se puedan desarrollar fácilmente en C. El objetivo de diseño iba a ser tan simple como fuera posible con un buen sonido y buenos gráficos. Bastante tiempo que deja suficientes recursos para ejecutar juegos interesantes. Se hizo hincapié porque es fácil y divertido de montar, para cualquier grupo de aficionados. El diseño final incluye dos fichas: una ATmega644 y un AD725 RGB a convertidor de NTSC. Permite además utilizar mandos de la *Nintendo SNES*.



**Figura 48.** Uzebox (fuente: <http://es.wikipedia.org/wiki/Uzebox>).

### 2.2.4. Fuzebox

La *Fuzebox* es una consola en formato de kit de automontaje y es una derivación de la consola *Uzebox*. El kit lo distribuye *Adafruit* y tiene las mismas características que la *Uzebox* sólo con la excepción de algunos cambios en el circuito eléctrico y en algunos casos los puertos de los mandos de la *SNES* pueden ser sustituidos por los *NES*.



**Figura 49.** *Fuzebox (fuente: <http://www.adafruit.com/blog/category/fuzebox/>).*



# CAPÍTULO 3:

## NORMATIVA Y

### ASPECTOS LEGALES

En este apartado se procede a explicar la diferente normativa a la cual está sujeto un diseño de estas características. Tanto para el *hardware* como para el *software*.

#### 3.1. Normativa referente al *hardware*

Toda la normativa referente al *hardware* tiene como objetivo el salvaguardar la integridad física del usuario, así como la del propio dispositivo y los dispositivos que están a su alrededor.

##### 3.1.1. Directiva 2002/95/CE (RoHS)

La directiva 2002/95/CE de restricción de sustancias peligrosas en dispositivos eléctricos y electrónicos, (*RoHS* del inglés "*Restriction of Hazardous Substances*"), fue adoptada en febrero de 2003 por la Unión Europea. Entró en vigor en julio de 2006.

Esta directiva restringe el uso de los siguientes materiales en la fabricación de varios tipos de equipos eléctricos y electrónicos, al ser considerados peligrosos:

- Plomo
- Mercurio
- Cadmio
- Cromo VI (Cromo hexavalente)
- PBB
- PBDE

Se establece que la cantidad de estas sustancias en el peso total de los componentes que conforman el dispositivo, no pueden superar la cantidad de 0,1% y para el Cadmio no pueden superar una cantidad de un 0,01%.

Esta directiva se impone los siguientes tipos de equipos:

- Electrodomésticos
- Equipos de comunicaciones e IT
- Aparatos eléctricos de consumo
- Dispositivos de iluminación, incluyendo las bombillas de filamento
- Herramientas eléctricas y electrónicas
- Juguetes, equipos deportivos y de recreo
- Máquinas expendedoras

Esta directiva afecta a la selección de materiales y componentes necesarios para la realización del proyecto, haciendo necesaria la comprobación de que los materiales y componentes utilizados cumplan la directiva.

### *3.1.2. Compatibilidad electromagnética*

La compatibilidad electromagnética (CEM o EMC) se define como la capacidad de cualquier aparato, equipo o sistema para funcionar de forma satisfactoria en su entorno electromagnético sin provocar perturbaciones electromagnéticas sobre el entorno.

Por tanto, se diferencian los dispositivos, aparatos o sistemas participantes en un entorno electromagnético según dos grupos:

- Emisor: produce tensiones, intensidades o campos electromagnéticos que, potencialmente, son la causa de perturbaciones a otros elementos del entorno, incluidos ellos mismos.
- Susceptible: su uso puede verse degradado en presencia de perturbaciones electromagnéticas, las cuales provocan otros equipos o ellos mismos.

Además, dado que un sistema puede estar compuesto por subsistemas, se tienen que estudiar las posibles interferencias internas entre ellos mismos.

La directiva 2004/108/CE establece dos límites a cumplir: el límite de emisión por los dispositivos considerados emisores, y el límite de inmunidad para los dispositivos considerados como susceptibles. Esta directiva ha sido actualizada en 2007 por los distintos países de la Unión Europea, haciendo estos límites más restrictivos.

### *3.1.3. Directiva 91/157/CEE (Baterías)*

Esta directiva se encarga de regular el uso de baterías o pilas que contienen un alto índice de sustancias peligrosas. Por ejemplo queda limitado el uso de mercurio en pilas de manganeso y el contenido de materiales pesados en ellas. También hace especial énfasis a la gestión de éste tipo de accesorio como residuo donde aparte de la reducción de

sustancias tóxicas para las personas, la directiva está orientada a la mejora del reciclaje de estos productos o a la mayor efectividad de los tratamientos que deben ser sometidos una vez ha finalizado su vida útil.

Respecto al proyecto, se pretende utilizar un alimentador externo, pero en el caso de implementar posteriormente la mejora de incluir un sistema de pilas o baterías, se debería cumplir esta directiva.

#### 3.1.4. Directiva 2002/96/CE (WEEE)

Esta directiva se encarga de que los países de la Unión Europea implanten una legislación nacional para los fabricantes e importadores de aparatos eléctricos y electrónicos.

El objetivo de esta medida es que tanto fabricantes como importadores se responsabilicen de los costes de la recuperación de sus productos como residuos que deben recibir un reciclaje o tratamiento especial.

Esta medida es necesaria ya que los componentes electrónicos o eléctricos deben ser tratados de forma especial cuando finalizan su vida útil, ya que no pueden someterse a tratamientos de destrucción de residuos tales como la incineración, puesto que la emisiones resultantes serían altamente contaminantes, llegando incluso a afectar a la salud humana.

Dado que este proyecto no pretende ser fabricado con el objetivo de su distribución, si se decide desecharlo, el dispositivo deberá ser entregado en los puntos verdes habilitados para tal fin.

En el caso de que se pretenda distribuir este producto, ha de tenerse en cuenta esta directiva y cumplirla adecuadamente.

## 3.2. Normativa referente al *software*

La normativa que hace referencia al *software* hace referencia principalmente a las leyes de la propiedad intelectual, y a las licencias de uso de los programas utilizados.

#### 3.2.1. Copyright

El juego diseñado está inspirado en el *Pong* de *Atari*, que fue desarrollado en 1972, y no tuvo copyright dado que la circuitería utilizada en aquella época no era reproducible, hubo numerosas máquinas que incluían diversas variaciones del mismo juego original.

Así pues al no disponer de *copyright*, el proyecto realizado no vulnera ningún tipo de derecho de autor.

### 3.2.2. *Licencias de uso del software utilizado*

Durante la realización del diseño de este proyecto, se han utilizado un gran número de herramientas de desarrollo, siendo necesaria en algunos casos la obtención de la licencia de uso correspondiente. En concreto se han utilizado los siguientes programas:

- El *CCS C*, de la compañía *CCS (Custom Computer Services)*, es un *software* que proporciona un completo entorno de desarrollo para trabajar con los microcontroladores PIC de *Microchip*. Las herramientas de desarrollo ofrecidas por *CCS* incluyen un compilador en C optimizado, con un depurador de errores y una gran cantidad de librerías que permiten facilitar el uso de este tipo de dispositivos. El uso de este *software* requiere de una licencia que se puede adquirir en su página web.
- *Proteus* es una compilación de programas de diseño y simulación electrónica, desarrollado por *Labcenter Electronics*, que consta de dos programas principales: *Ares* e *Isis*, y los módulos *VSM* y *Electra*. Para el uso de cualquiera de sus programas se necesita disponer de la licencia, que está disponible para su venta en la página web del desarrollador.
  - El programa *Isis* permite diseñar el esquema eléctrico del circuito que se desea realizar con componentes muy variados, desde resistencias hasta microcontroladores. Los diseños realizados en *Isis* pueden ser simulados en tiempo real, mediante el módulo *VSM*, asociado directamente con *Isis*.
  - *Ares* se encarga de enrutar, ubicar y editar componentes, y se utiliza para el diseño y fabricación de placas de circuito impreso, permitiendo importar un diseño procedente de *Isis*.
- *OrCAD* también es un programa que se usa para el diseño, simulación y delineación del hardware. También requiere la adquisición de una licencia para su uso, sin embargo existe una versión gratuita dirigida a los estudiantes que pese a tener ciertas limitaciones con respecto a la versión completa, sirve perfectamente para los propósitos de este proyecto.
- El programa *Bitmap2LCD*, permite la obtención del código C correspondiente a una imagen prediseñada por el usuario, facilitando el uso de visualizadores *LCD* gráficos. También requiere de una licencia de uso que se puede comprar en la página web del producto.
- Los programas de edición de texto e imagen utilizados durante el desarrollo de la memoria del proyecto pertenecen al paquete de *Microsoft Office 2010* y también requieren de una licencia de uso.



# CAPÍTULO 4: INGENIERÍA DE CONCEPCIÓN

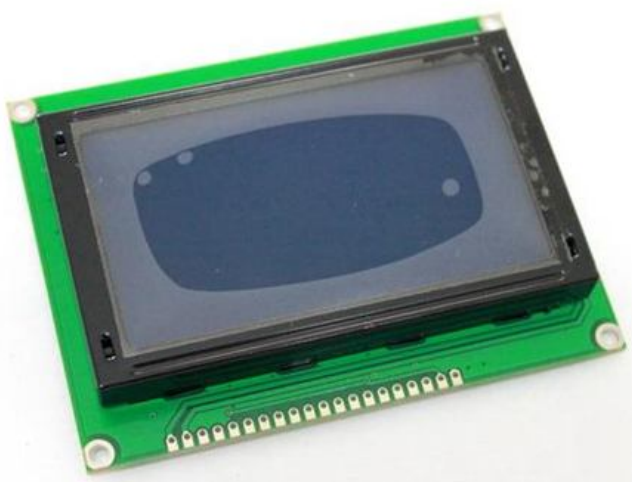
En este capítulo se expone la propuesta inicial del proyecto, tanto *hardware* como *software*.

## 4.1. Solución propuesta

A continuación se detallará la solución escogida para desarrollar este proyecto.

### 4.1.1. Visualización

Para la visualización del videojuego, se ha elegido utilizar un *LCD* gráfico de 128x64 píxeles, que permita ver en pantalla el desarrollo del videojuego.



**Figura 50.** *LCD gráfico (fuente: <http://www.lelong.com.my/128x64-dots-graphic-lcd-module-jhd12864e-blue-backlight-104439589-2013-03-Sale-P.htm>).*

Se ha optado por esta solución debido a que permite realizar un juego con una buena resolución, pudiendo además ser conectado directamente al microcontrolador al disponer el *LCD* de un controlador interno para este

propósito. El acondicionamiento de señal que necesita un *LCD* para ser utilizado por un microcontrolador varía en función del modelo, pero es prácticamente nulo, hecho que resulta en un esquema eléctrico más sencillo. Otra ventaja de este tipo de visualización, es que su consumo eléctrico es menor respecto al de sus alternativas, permitiendo incluir una fuente de alimentación de menor valor o dotar al sistema de otros dispositivos.

#### 4.1.2. Microcontrolador

La elección del microcontrolador es la parte más crítica del proyecto, puesto que es el encargado de ejecutar todas las instrucciones descritas en el programa realizado.

Después de una amplia comparativa se ha decidido utilizar como microcontrolador el modelo PIC18F4550 de *Microchip*, ya que sus prestaciones son las adecuadas de acuerdo a las necesidades de este proyecto. El aspecto más crítico considerado en cuanto a su elección ha sido principalmente el número de puertos, y sobretodo la cantidad de memoria de programa disponible y memoria RAM. Al tener que controlar un *GLCD* el microcontrolador debe dedicarle prácticamente 2 puertos enteros, limitando pues el número de puertos restantes para las entradas y demás salidas. Se ha considerado que para el objetivo del diseño, son necesarios 4 puertos como mínimo, que son los que tiene el PIC18F4550.

En cuanto a la memoria de programa, el uso de *GLCDs* permite mostrar imágenes directamente guardadas en la memoria, la pega es que cada imagen pesa 1Kbyte (suponiendo un visualizador de 128x64 pixeles), así pues se precisa un dispositivo con una memoria elevada para poder almacenar un pequeño número de imágenes más el resto del programa. A su vez el dispositivo ha de tener más de 1Kbyte de memoria RAM disponible para poder utilizar el modo *fast* de visualización del *GLCD*, que permite una visualización mucho más dinámica y rápida. Por lo tanto los requisitos en cuanto a memoria son de al menos 2 Kbytes de RAM y 32 Kbytes de memoria de programa, que son precisamente los que tiene el PIC18F4550, de ahí su elección.

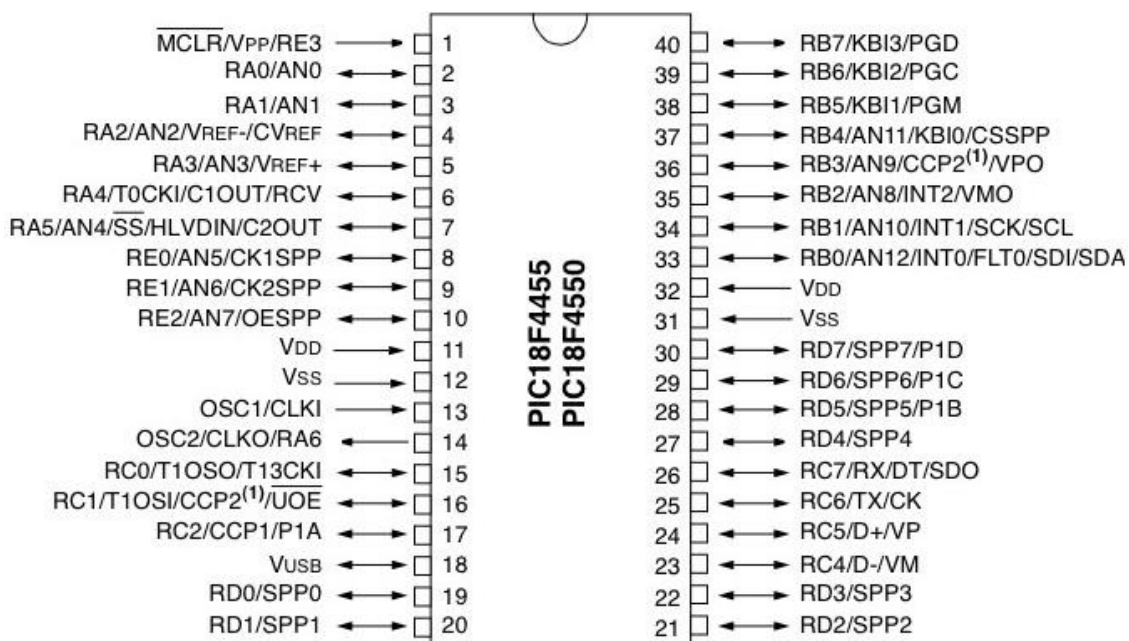
El PIC 18F4550, pertenece a la familia de microcontroladores PIC18 de gama alta. Posee una arquitectura *RISC* (*reduced instruction set computer*) de 16 bits longitud de instrucciones y 8 bits de datos. La siguiente tabla muestra en resumen las características fundamentales de este microcontrolador y de sus antecesores los PIC18F2455/2550/4455.

**Tabla 1.** Características del PIC18F4550 y sus antecesores.

CARACTERÍSTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F4550
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24.576	32.768	24.576	32.768

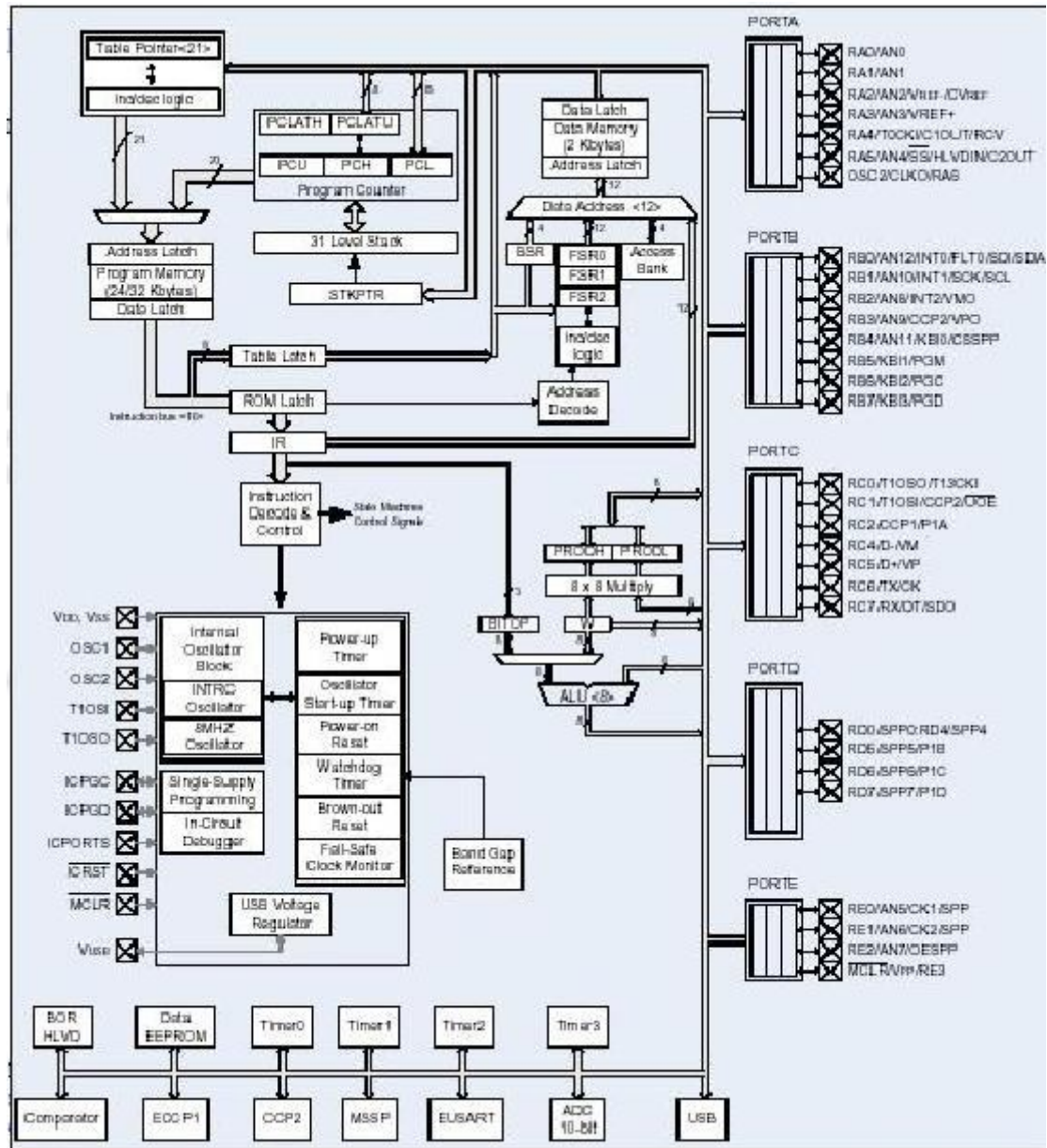
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de E/S	24	24	35	35
Temporizadores	4	4	4	4
Módulos de Comparación/Captura/PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura/PWM mejorado (ECCP)	0	0	1	1
Canales de Comunicación Serie	MSSP.EUSART	MSSP.EUSART	MSSP.EUSAR	MSSP.EUSAR
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Canales de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)
Encapsulados	PDIP28 pines SOIC28 pines	PDIP28 pines SOIC28 pines	PDIP40 pines QFN 40 pines TQFP 40 pines	PDIP40 pines QFN 40 pines TQFP 40 pines

En la siguiente figura se muestra el patillaje del microcontrolador, en un encapsulado del tipo DIP40, que es el que se utilizará en el montaje.



**Figura 51.** Pines del PIC18F4550 (fuente: <http://www.microchip.com>).

También se procede a mostrar un diagrama de bloques del microcontrolador en cuestión, distribuido por su fabricante *Microchip*.



**Figura 52.** Diagrama de bloques del PIC18F4550 (fuente: <http://www.microchip.com>).

#### 4.1.3. Mandos

El videojuego requiere de la interacción humana para su funcionamiento, por tanto se ha optado por incluir dos pulsadores que servirán para que el jugador pueda tener un control directo sobre el videojuego.



**Figura 53.** Pulsador normalmente abierto utilizado para el control del videojuego (fuente: <http://www.diotronic.com>).

Este tipo de pulsadores presentan el inconveniente de generar rebotes al ser apretados, por tanto se deberá implementar un filtro pasa bajos en el sistema que elimine este ruido indeseado.

#### 4.1.4. Plataforma de desarrollo

Para el correcto diseño del sistema, hacen falta utilizar diversas herramientas de desarrollo para poder realizar los procesos de programación, compilación y depuración del código, así como su correcta simulación.

Por este motivo se deciden seleccionar los siguientes programas para poder realizar estas funciones:

- El *CCS C* permite compilar y depurar el código de programa, así como editarlo tantas veces como se desee. Además cuenta con un gran número de librerías y ejemplos que pueden servir de ayuda puntual en ciertos aspectos del diseño.
- El *Isis* del entorno *Proteus* permite diseñar esquemas electrónicos con microprocesadores, así como simular en tiempo real el funcionamiento del código.

## 4.2. Esquemas iniciales

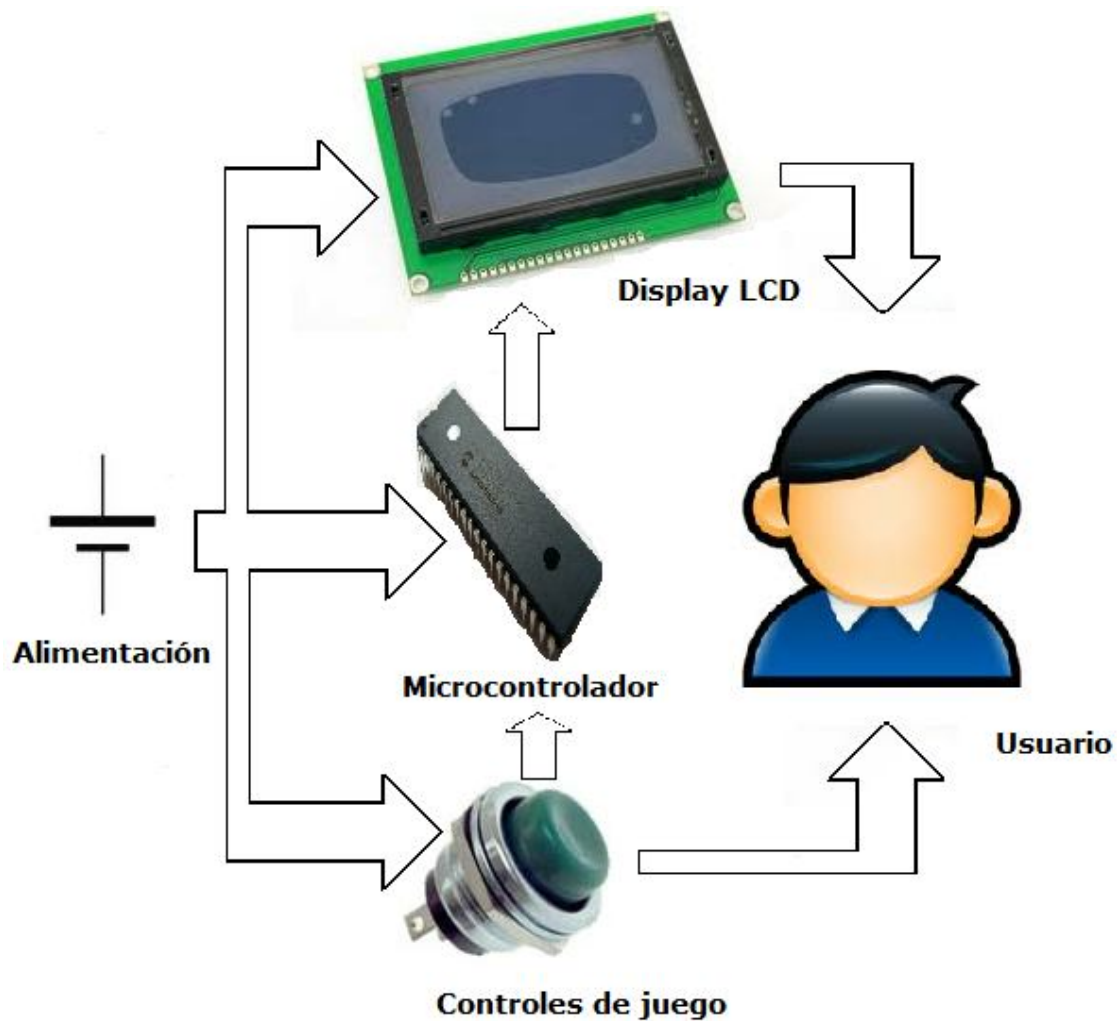
El esquema inicial se basa en un microcontrolador, alimentado externamente, que tiene como entradas de datos dos pulsadores, y como salida un visualizador *LCD*.

Así pues se pueden distinguir los siguientes bloques en este sistema:

- Alimentación: Implementada de manera externa.
- Control: Implementado con un microcontrolador.
- Dispositivos de entrada de datos: Implementado con pulsadores.
- Dispositivos de salida de datos: Implementado con el *display LCD*.

Por último, el encargado de realimentar el sistema, es decir, de actuar en las entradas en función de las salidas, no es otro que el jugador. Así pues el jugador también es parte del sistema.

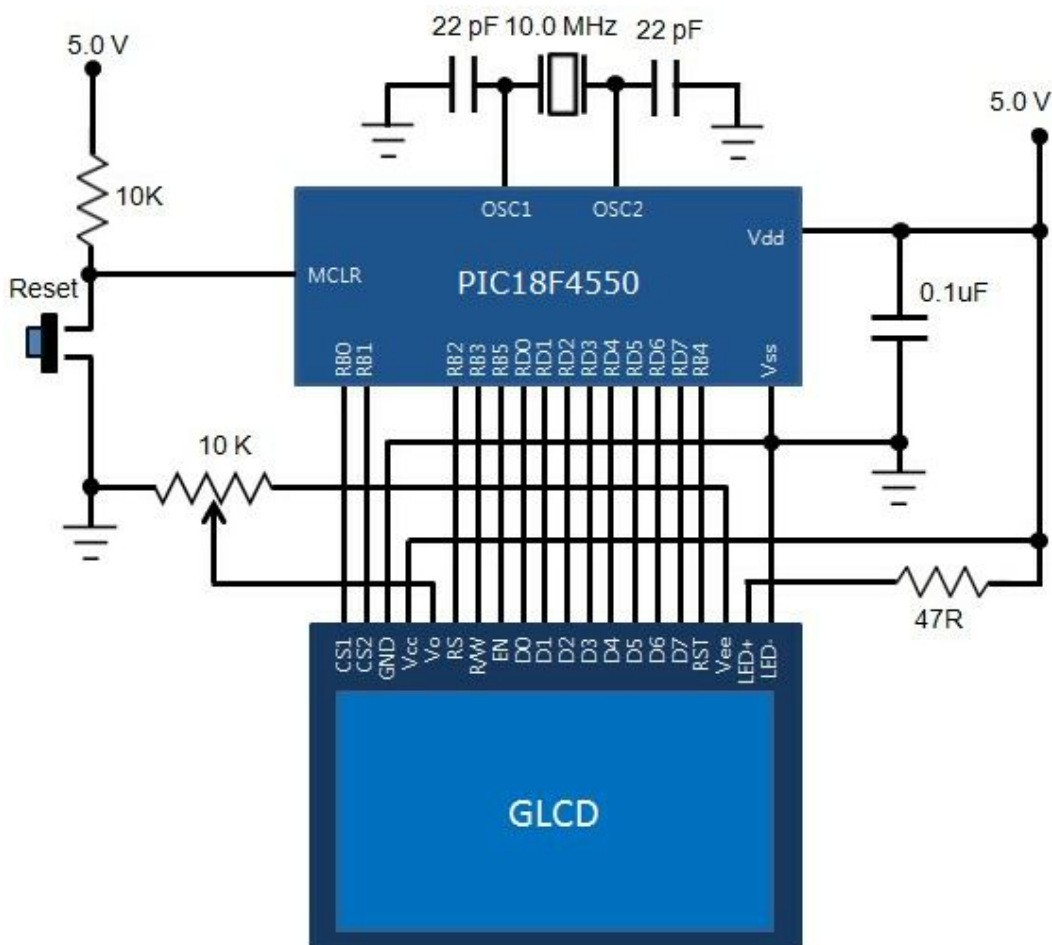
Seguidamente se muestra un diagrama de bloques inicial del sistema.



**Figura 54.** Diagrama de bloques del sistema (fuente: propia).

#### 4.2.1. Visualización

La conexión típica entre un *LCD* gráfico y un microcontrolador PIC consiste en dedicar todo un puerto del microcontrolador al envío de datos llevándolos directamente al *LCD*. A parte de este puerto, el *LCD* también necesita de diversas señales de control procedentes también desde el microcontrolador, así que en total serán 14 las conexiones entre *LCD* y microcontrolador. El *LCD* también necesita alimentación externa y además incluye un control de brillo implementado con un potenciómetro. Seguidamente se muestra un esquema de conexión típico entre un *GLCD* y un microcontrolador PIC.



**Figura 55.** Esquema de conexión típico de un GLCD (fuente: propia).

#### 4.2.2. Controles del videojuego

El juego incluye dos pulsadores, que irán conectados directamente a un puerto del microcontrolador, dichos pulsadores contarán además con un sistema de filtrado que elimine los posibles rebotes característicos de estos componentes.

Además se incluye un tercer botón, que será el encargado de realizar un *reset* a la máquina.

#### 4.2.3. Microcontrolador

El conexionado restante del microcontrolador solo incluye la conexión de un reloj externo que proporcione la frecuencia de trabajo al sistema. Este esquema de conexión viene dado por el fabricante en la hoja de características técnicas del microcontrolador.

## 4.3. Juego a implementar

Se pretende realizar el diseño de un videojuego inspirado en el mítico *Pong* de *Atari*. Pese a que el juego original estaba pensado para ser un simulador del tenis de mesa que enfrentaba a dos jugadores, la versión que se va a implementar es para un solo jugador. Se pretende crear un juego que de habilidad que basado en *Pong*, imite a un jugador de tenis de mesa. Hay un cambio de paradigma respecto al juego original, y es que en esta versión no se trata de vencer al rival, sino de evitar ser derrotados. Otra novedad es que solo veremos nuestro campo y que esta vez jugaremos en vertical.

El objetivo del juego es aguantar el máximo tiempo posible devolviendo el balón, entonces, al llegar a cierta puntuación, se subirá de nivel y consecuentemente de dificultad. El juego constará de 5 niveles, cada uno de ellos más exigente que el anterior, y si se logra finalizar el quinto nivel, se gana la partida.

Dispondremos de 5 intentos para conseguirlo, reapareciendo tras cada fallo en el inicio del nivel en el que nos encontramos. Si se es derrotado, la máquina guarda la puntuación y te invita a iniciar una nueva partida. El hecho de guardar la puntuación fomenta la competición entre usuarios por tal de lograr la máxima puntuación.

Se ha decidido llamar al juego *Deadly Pong* (en castellano *Pong Mortal*), debido a la influencia de *Pong* y al carácter del videojuego que hace que el jugador trate de resistir (o sobrevivir) todo lo que pueda en pantalla para ganar. En caso de no conseguirlo el jugador pierde una vida, por eso se ha utilizado el adjetivo "mortal".

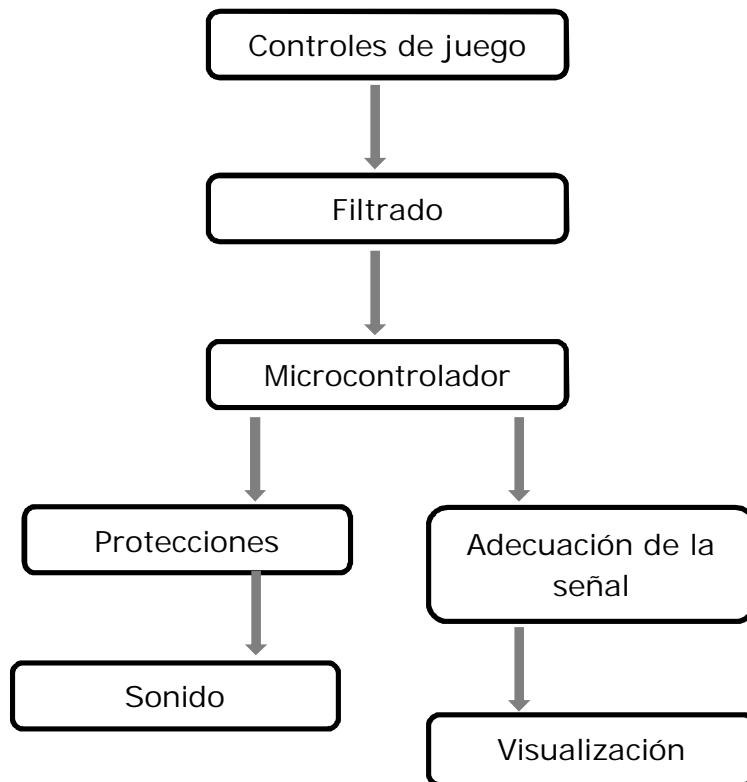


# CAPÍTULO 5: INGENIERÍA DE DETALLE

En este capítulo se expone el diseño final realizado, tanto *hardware* como *software*, así como su desarrollo y las herramientas utilizadas.

## 5.1. Esquemas de detalle del *hardware*

El hardware de este proyecto está compuesto por los bloques principales, encargados del control, interacción, visualización y sonido, y por bloques secundarios que hacen funciones de filtrado, adecuación de la señal, y protección. El conexionado entre los bloques mencionados se muestra más abajo.



**Figura 56.** Esquema de bloques del diseño final del videojuego (fuente: propia).

Seguidamente se va a proceder a explicar cada uno de los bloques que conforman el juego.

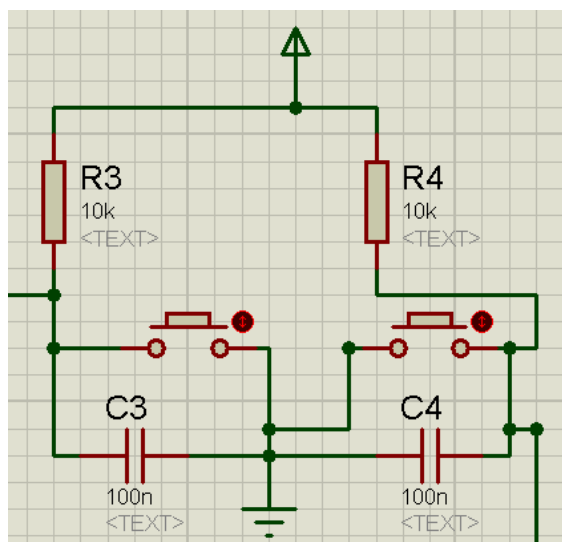
#### 5.1.1. Controles de juego

El juego se controla mediante 3 pulsadores conectados al microcontrolador a través de la etapa de filtrado. Uno de los pulsadores ocupa el terminal de reset y los otros dos van al terminal RA0 (botón derecho) y al RA1 (botón izquierdo).

#### 5.1.2. Filtrado

El bloque de filtrado se encarga de eliminar las pulsaciones no deseadas producidas en los rebotes ocurridos al pulsar los botones.

La célula de filtrado está compuesta por una red RC que actúa de filtro pasa bajos, en la siguiente figura se puede observar esta célula formada por las resistencias y los condensadores  $R_3$  y  $C_3$ , y  $R_4$  y  $C_4$ , respectivamente.

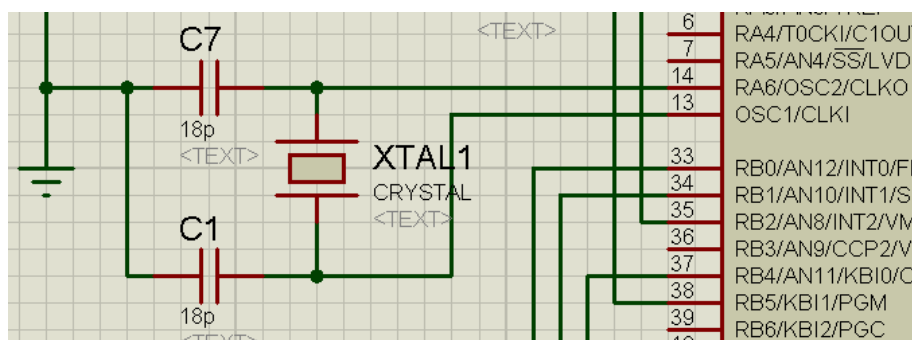


**Figura 57.** Esquema de los pulsadores con los filtros (fuente: propia).

### 5.1.3. Microcontrolador

El dispositivo de control principal es el microcontrolador PIC18F4550, de la marca *Microchip*. Los microcontroladores integran tanto un microprocesador como periféricos típicos, por tanto, apenas necesitan circuitería externa.

El único circuito externo conectado al microcontrolador en este diseño, es el del cristal de cuarzo de 12 MHz, encargado de ser el reloj del circuito. El conexionado de este cristal se realiza siguiendo el esquema descrito por el fabricante del microcontrolador en su hoja de características técnicas.



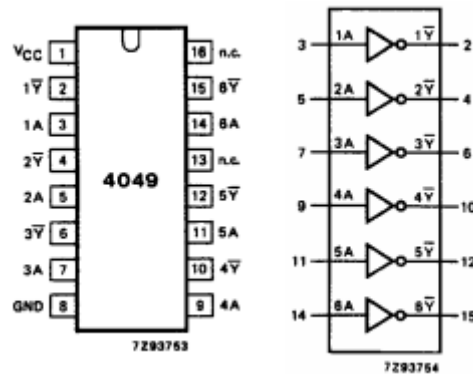
**Figura 58.** Esquema de conexión del cristal de cuarzo al microcontrolador (fuente: propia).

### 5.1.4. Adecuación de la señal

Este bloque se encarga principalmente de adecuar la señal de salida del microcontrolador para que pueda actuar sobre el *LCD*. Esta necesidad es debida al uso de una de las librerías del programa *CCS C*, que habilita el interactuar con un *LCD* gráfico, que está dirigida a un modelo de *LCD* específico que presenta la diferencia respecto al utilizado en este proyecto de que sus entradas CS1 y CS2 no están negadas. Por tanto para poder

utilizar esta librería con el JHD12864E, se debe incorporar al sistema de un driver que niegue estas señales antes de llegar al *LCD*.

Para este propósito se ha seleccionado el circuito integrado 74HC4049, que cumple perfectamente con los requisitos demandados.



**Figura 59.** Diagrama de pines y símbolos lógicos del 74HC4049 (fuente: <http://www.datasheetcatalog.com>).

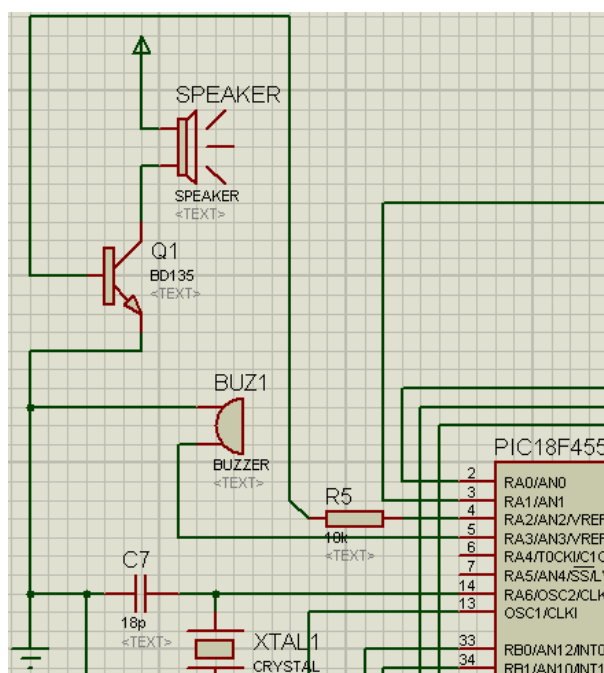
#### 5.1.5. Protecciones

Para evitar corrientes demasiado altas en el altavoz, se ha optado por incluir un transistor modelo BD135 para controlar la corriente que pasa por el altavoz, conectando en su base una resistencia de valor óhmico elevado. Sabiendo pues el valor de la amplificación típica de este modelo de transistor, solo hay que ajustar la resistencia de base a un valor que nos proporcione el nivel de potencia deseado en el altavoz.

#### 5.1.6. Sonido

Se han incluido en el diseño dos fuentes de sonido, un zumbador piezocerámico que oscila a 3 MHz encargado de reproducir los rebotes de la bola, y un pequeño altavoz de 8 ohm encargado de reproducir las melodías del juego.

El conexionado de los mismos se ha realizado de la siguiente manera:

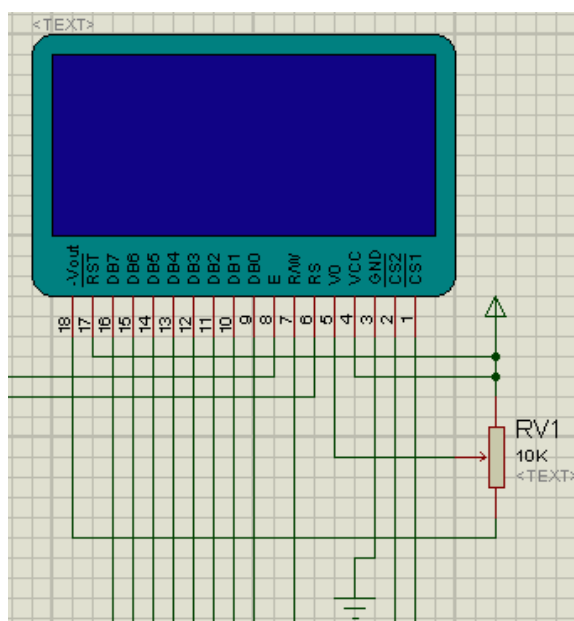


**Figura 60.** Esquema del zumbador y el altavoz junto con las protecciones (fuente: propia).

#### 5.1.7. Visualización

El bloque de visualización se basa principalmente en el mostrado en el apartado de ingeniería de la concepción. Se conectan los pines de datos y control al PIC (dos pasan por el bloque de adecuación de la señal) y se conectan dos pines a un potenciómetro para regular el brillo de la pantalla como se explicará más adelante.

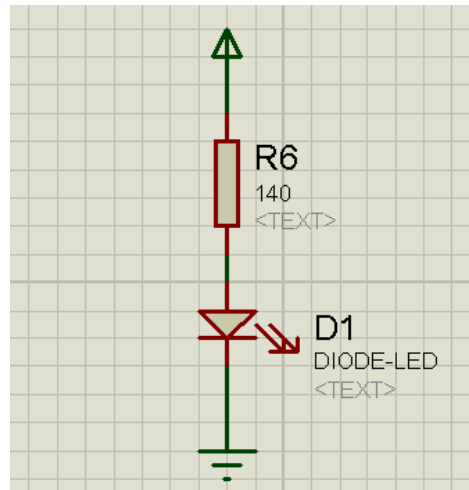
Como visualizador *LCD* gráfico se ha utilizado el JHD12864E, un dispositivo de altas prestaciones y una buena relación calidad.



**Figura 61.** Esquema del control de brillo del LCD (fuente: propia).

### 5.1.8. Otros

Se ha dotado al dispositivo de un interruptor maestro que encienda y apague el sistema desconectándolo de la alimentación. Para saber cuando está conectado, se incluye en el sistema un LED que se iluminará siempre y cuando el sistema esté alimentado.

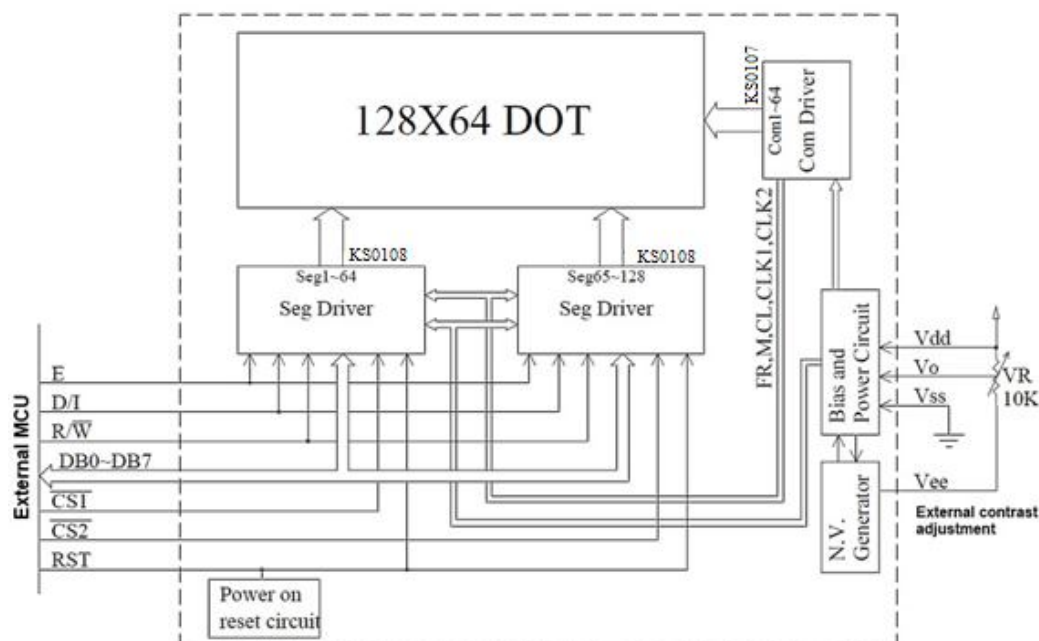


**Figura 62.** Esquema del indicador LED (fuente: propia).

## 5.2. Funcionamiento de un LCD gráfico

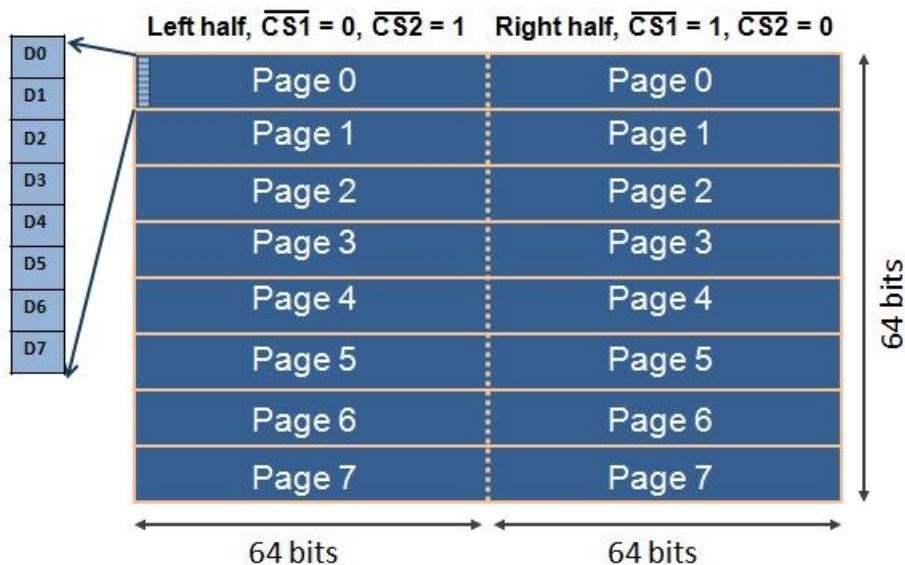
El uso de un LCD gráfico, o GLCD, cambia drásticamente el aspecto del proyecto dándole un toque más profesional. Su uso proporciona la libertad para representar datos de la que no disponen los dispositivos LCD convencionales, proporcionando una resolución difícilmente alcanzable por otras alternativas de visualización como pueden ser las matrices de LEDs.

El GLCD seleccionado para el desarrollo de este proyecto es el JHD12864E, el cual es tiene una pantalla de 128x64 píxeles monocromática. Usa dos tipos de controladores integrados modelos KS0107 y KS0108 fabricado por Samsung. El GLCD incorpora dos drivers KS0108 encargados de controlar 64 columnas cada uno de la matriz de puntos del GLCD. El controlador KS0107 es un driver de 64 canales que genera la señal de tiempos para controlar a los dos drivers KS0108. Ambos controladores son muy populares y han sido implementados por gran cantidad de fabricantes de GLCDs.



**Figura 63.** Diagrama de bloques de los controladores KS0107 y KS0108 implementados en un módulo GLCD de 128x64 (fuente: <http://embedded-lab.com/blog/?p=2398>).

El KS0107 maneja las 64 líneas del visualizador, COM1-COM64. El primer KS0108 se encarga de la primera mitad de las columnas (SEG1-SEG64) y la segunda mitad (SEG65-SEG128) son controladas por el segundo controlador KS0108. Se puede acceder individualmente a cada una de las dos mitades del visualizador utilizando los pines de selección de chip de cada controlador KS0108, CS1 y CS2 respectivamente. Cada mitad se divide en 8 páginas horizontales (0-7) de 8 bits de altura cada una.



**Figura 64.** Esquema de un GLCD de 128x64 (fuente: <http://embedded-lab.com/blog/?p=2398>).

Empezando desde la página 0 en la mitad izquierda (/CS1=0) si se transmite un byte de datos, este aparecerá en la primera columna de la página 0. Si se repite esta acción 64 veces, se cambia a la segunda mitad del visualizador (/CS2=0) y se repite la acción hasta la posición 128, las primeras 8 líneas del visualizador habrán sido graficadas. Para las 8 líneas siguientes, se procede de igual manera solo que cambiando a la dirección de página 1. Por lo tanto el conjunto de bytes necesarios para completar toda la pantalla es de  $2 * 64 \text{ pixeles} * 8 \text{ bits} = 1024 \text{ bytes}$ .

El módulo del *GLCD* incorpora un generador interno de voltaje negativo que proporciona una tensión negativa en el pin  $V_{EE}$ . Al sistema normalmente se le añade un potenciómetro de  $10K\Omega$  que es conectado entre los pines de  $V_{CC}$  y  $V_{EE}$  para fijar el nivel de tensión de trabajo del *LCD* (contraste) en el pin  $V_0$ .

PIN NO.	SYMBOL	DESCRIPTION	FUNCTION
1	VSS	GROUND	0V (GND)
2	VDD	POWER SUPPLY FOR LOGIC CIRCUIT	+5V
3	V0	LCD CONTRAST ADJUSTMENT	
4	RS	INSTRUCTION/DATA REGISTER SELECTION	RS = 0 : INSTRUCTION REGISTER RS = 1 : DATA REGISTER
5	R/W	READ/WRITE SELECTION	R/W = 0 : REGISTER WRITE R/W = 1 : REGISTER READ
6	E	ENABLE SIGNAL	
7	DB0	DATA INPUT/OUTPUT LINES	8 BIT: DB0-DB7
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	CS1	CHIP SELECTION	CS1=1,CHIP SELECT SIGNAL FOR IC1
16	CS2	CHIP SELECTION	CS2=1,CHIP SELECT SIGNAL FOR IC2
17	RST	RESET SIGNAL	RSTB=0,DISPLAY OFF,DISPLAY FROM LINE 0.
18	VEE	NEGATIVE VOLTAGE FOR LCD DRIVING	-10V
19	LED+	SUPPLY VOLTAGE FOR LED+	+5V
20	LED-	SUPPLY VOLTAGE FOR LED-	0V

**Figura 65.** Conexiones del JHD12864E (fuente: datasheet JHD12864E).



Pese a que muchos dispositivos comparten este tipo de controlador, su conexión no está estandarizada, por lo que se debe consultar el *datasheet* del fabricante para conocer como están distribuidos los pines en el dispositivo.

El KS0107/KS0108 no posee un generador de caracteres, por lo que se deberá implementar en el *software* del microcontrolador. Estos controladores soportan un gran número de instrucciones, las cuales se resumen en la figura inferior. Nótese que el pin RS solo está en estado alto durante las operaciones de lectura y escritura, y permanece en estado bajo cuando el byte transmitido es una instrucción.

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display on/off	L	L	L	L	H	H	H	H	H	L/H	Controls the display on or off. Internal status and display RAM data is not affected. L: OFF, H: ON
Set address (Y address)	L	L	L	H	Y address (0 - 63)						Sets the Y address in the Y address counter.
Set page (X address)	L	L	H	L	H	H	H	Page (0 - 7)			Sets the X address at the X address register.
Display start line (Z address)	L	L	H	H	Display start line (0 - 63)						Indicates the display data RAM displayed at the top of the screen.
Status read	L	H	Busy	L	On/Off	Reset	L	L	L	L	Read status. BUSY L: Ready H: In operation ON/OFF L: Display ON H: Display OFF RESET L: Normal H: Reset
Write display data	H	L	Write data								Writes data (DB0:7) into display data RAM. After writing instruction, Y address is increased by 1 automatically.
Read display data	H	H	Read data								Reads data (DB0:7) from display data RAM to the data bus.

**Figura 66.** Set de instrucciones del JHD12864E (fuente: *datasheet* JHD12864E).

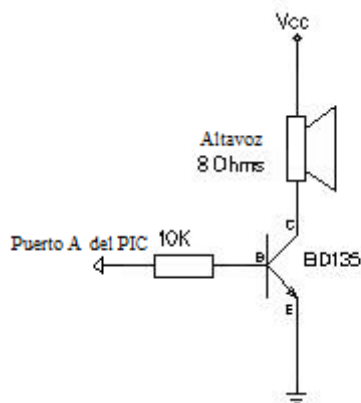
## 5.3. Selección de componentes y cálculos justificativos

Durante la realización del esquema final del prototipo, se han tenido que escoger valores de ciertos componentes según las características del diseño. Así pues se procede a justificar la elección de estos componentes.

### 5.3.1. Altavoz

Dado que se desea un sistema de sonido sencillo de pequeñas dimensiones, se tiene que ajustar la corriente que consume el altavoz mediante el transistor BD135 para que no disipe demasiada potencia.

Sabiendo que el esquema de conexión es la siguiente:



**Figura 67.** Esquema de conexión del altavoz al PIC (fuente: propia).

Y que el valor de la resistencia de base ha de ser elevado para forzar una corriente pequeña, por ejemplo de  $10\text{K}\Omega$ .

Se pueden establecer las siguientes ecuaciones:

Ecuación de la malla base-emisor

$$V_{PA} - i_B \cdot R_B - V_{BE} = 0 \quad (1)$$

Ecuación de la malla colector-emisor

$$V_{CC} - i_C \cdot R_{\text{Altavoz}} - V_{CE} = 0 \quad (2)$$

Relación entre intensidades

$$i_C = i_B \cdot \beta_{BD135} \quad (3)$$

Según el *datasheet* del transistor BD135, su  $V_{BE}$  típica en saturación es de aproximadamente de  $200\text{mV}$  y su ganancia en corriente continua ( $\beta$ ) mínima es 40. Por lo que se puede establecer que:

$$i_B = \frac{V_{PA} - V_{BE}}{R_B} = \frac{5 - 0,2}{10000} = 480 \mu A$$

$$i_C = i_B \cdot \beta_{BD135} = 480 \cdot 10^{-6} \cdot 40 = 19,2 \text{ mA}$$

Con lo que se puede establecer el valor de potencia máxima que disipará el altavoz.

$$P_{\text{Altavoz}} = i_C^2 \cdot R_{\text{Altavoz}} = 0,0192^2 \cdot 8 = 2,94 \text{ mW}$$

Teniendo en cuenta que esta sería la potencia que consumiría el altavoz si estuviese conectado en continua, y que en la práctica esté circuito está trabajado para trabajar en alterna, la potencia máxima total resultante será la mitad, en concreto 1,47mW. Pese a todo hay que tener cuidado al programar ya que si por error se dejase el transistor BD135 saturado, puede resultar en la degradación del dispositivo.

Finalmente solo queda seleccionar un altavoz que sea compatible con las especificaciones calculadas, y por ello se elige un altavoz de 0,3 W de potencia máxima admisible de la marca *FoneStar*, ya aguantará perfectamente la potencia disipada y además ofrece una solución sencilla al ser de tamaño reducido.

### 5.3.2. Resistencia LED

El sistema incluye un LED verde que indica cuando está conectada la alimentación. Puesto que el LED no puede conectarse directamente a alimentación debido a que la gran intensidad resultante destruiría el componente, se opta por conectar una resistencia en serie para minimizar esa corriente. Sabiendo que un LED verde tiene una caída de tensión de 2,2V y una corriente nominal de 20mA, se puede calcular la resistencia necesaria para conseguir estos valores.

$$\begin{aligned} V_{CC} - V_{LED} - I_{LED} \cdot R_{serie} &= 0 \\ R_{serie} &= \frac{V_{CC} - V_{LED}}{I_{LED}} = \frac{5 - 2,2}{0,02} = 140 \Omega \end{aligned} \quad (4)$$

Así pues la resistencia necesaria es de valor 140 ohmios (serie E48) y tiene una tolerancia del 2%.

### 5.3.3. Zumbador

El otro dispositivo de sonido que incorpora el sistema es un zumbador encargado de hacer los efectos de sonido correspondientes al rebote de la bola. Puesto que el zumbador irá conectado directamente a un pin del puerto A del microcontrolador, se tiene que seleccionar uno que pueda ser

alimentado por esa corriente y voltaje, dado que el PIC18F4550 solo puede proporcionar una intensidad de 25mA por pin y 5V.

Por lo tanto se ha escogido un modelo de zumbador piezocerámico que puede trabajar entre los 3 y los 24 voltios y con una corriente de 10mA.

#### 5.3.4. Condensadores de los pulsadores

Para corregir el efecto rebote en los pulsadores, se añade al sistema unos condensadores que actúen como filtro pasabajos, eliminando el ruido producido por este fenómeno.

Para calcular el valor de los condensadores hay que tener en cuenta que se desea un filtrado de al menos un 1 ms que es lo aconsejable para evitar el molesto efecto rebote. Sabiendo que el valor de la resistencia escogida es de  $10K\Omega$ , se calcula el valor del condensador:

$$t = R \cdot C$$
$$C = \frac{t}{R} = \frac{0,001}{10000} = 100nF \quad (5)$$

Por lo tanto el valor de los condensadores utilizados como filtro será de 100nF.

## 5.4. Diseño del *software*

Se procede a explicar los detalles de programación del juego, y sus funciones principales. La programación del videojuego ha partido de la base de las librerías dadas por el programa *CCS C*, que poseen rutinas para trabajar con *LCDs* gráficos y para reproducir tonos a través del PIC. El lenguaje de programación empleado para realizar el videojuego ha sido el C.

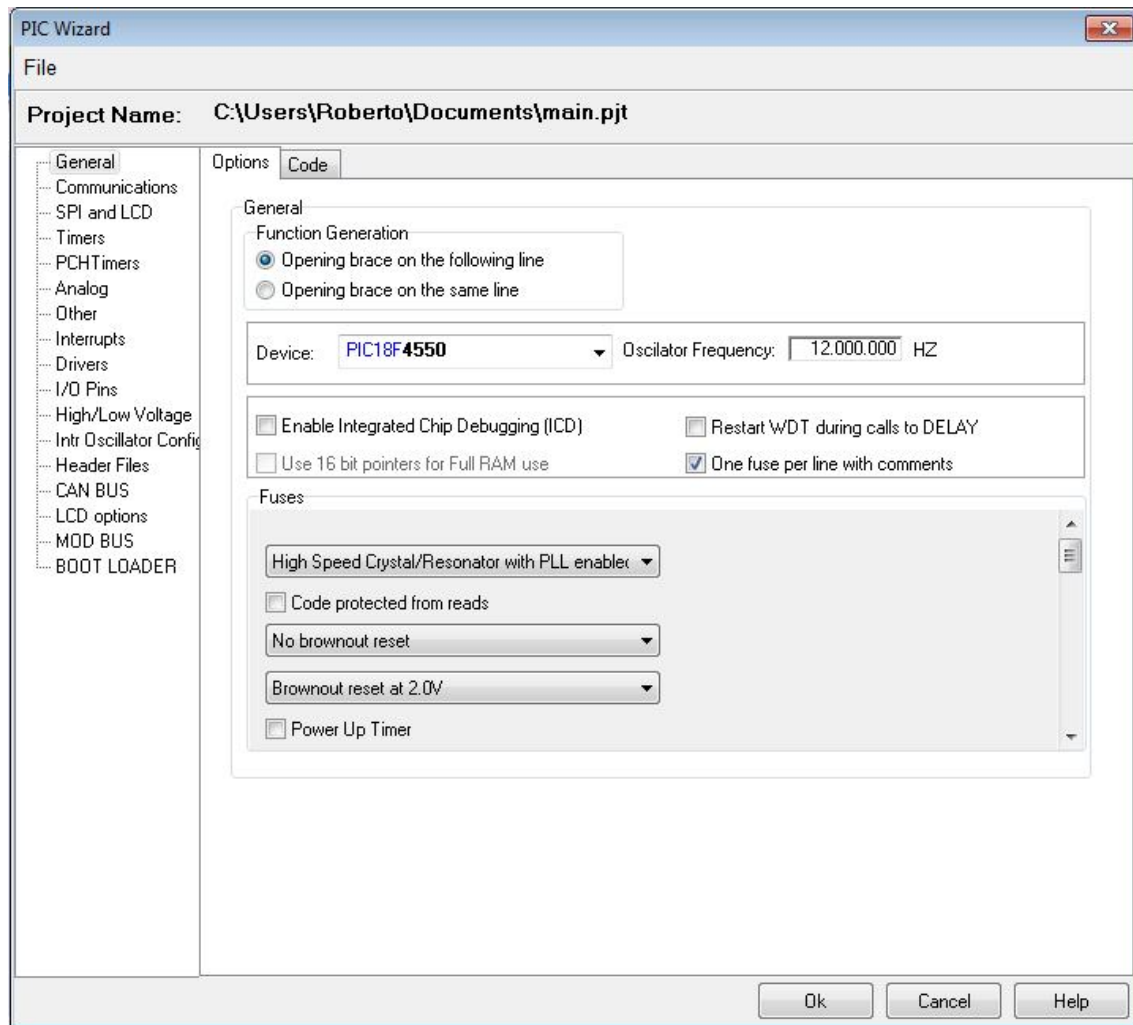
### 5.4.1. Entorno del *CCS C*

Para la programación y compilación del juego se ha utilizado el software *CCS C* un compilador que genera un código maquina eficiente, permite integrarse con *Proteus* para depurar y tiene una biblioteca completa de funciones pre-compiladas además de ser uno de los más utilizados junto a *MPLAB*. Adicionalmente se encuentra gran cantidad de información tanto en libros como en internet sobre este programa.

Para empezar la programación se debe empezar un nuevo proyecto. Para ello *CCS* tiene un asistente para configurar inicialmente el PIC a utilizar y algunas características del proyecto mediante el *PIC Wizard*. Para ello hay que ejecutar el programa y en la pestaña de *Project* seleccionamos *PIC Wizard*.

El programa requerirá guardar el proyecto y ponerle un nombre para poder continuar con el asistente, y así poder seleccionar las opciones de configuración del PIC a programar.

En la pantalla inicial del *PIC Wizard*, se deberá seleccionar el PIC a utilizar y la frecuencia y tipo de reloj con los que se vaya a trabajar. En la figura anterior se muestra la selección del microcontrolador para este proyecto, el PIC 18F4550 y una frecuencia de oscilación de 12 MHz.

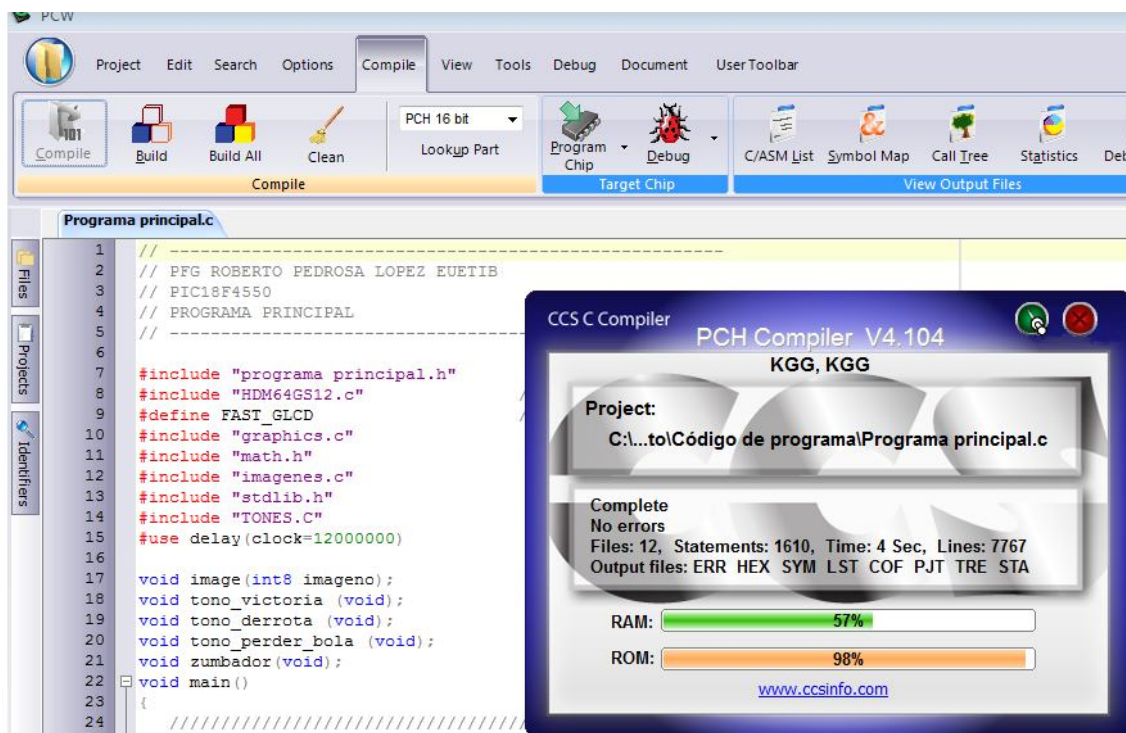


**Figura 68.** Project Wizard del entorno CCS C (fuente: propia).

Además esta herramienta permite configurar una gran cantidad de valores tales como definir el valor de los pines ya sean entradas o salidas, definir en los puertos aptos entradas analógicas, habilitar las interrupciones externas del puerto B, habilitar contadores y temporizadores, e incluso añadir al proyecto alguna de las librerías distribuidas por CCS. En el caso de este proyecto se configuran las entradas y salidas de acuerdo con el esquema final del diseño.

Una vez finalizado el asistente, ya se puede empezar a programar en el entorno de programación, para ello se añadirá automáticamente parte del código configurado previamente. Una vez realizado el programa, se puede compilar desde la pestaña *Compile* generando varios archivos en la carpeta raíz del proyecto, entre ellos el *.HEX*, que es el que tiene que ser grabado posteriormente sobre el PIC.

Así si el código de programa es correcto, la compilación será exitosa y se mostrará en pantalla la cantidad de memoria ROM y la RAM de nuestro dispositivo que ha ocupado nuestro código.



**Figura 69.** Captura del proyecto compilado con el entorno CCS C (fuente: propia).

#### 5.4.2. Librerías del CCS C

Como se ha destacado anteriormente, el entorno de programación CCS C dispone de una gran cantidad de librerías que ayudan en gran medida al desarrollo de cualquier aplicación, sirviendo de ayuda y ejemplo. Entre ellas se encuentran desde librerías especializadas en el uso de cada modelo de PIC, a diseñadas para facilitar el control de diversos dispositivos electrónicos tales como *LCDs*, memorias externas, sensores, etc. Incluso se dispone de librerías capaces de generar tonos musicales.

Para incluir el uso de cualquier librería en el programa desarrollado, simplemente tiene que utilizarse la siguiente expresión:

```
#include "nombre de la librería"
```

Durante el desarrollo de este proyecto se han utilizado las siguientes librerías:

- *18F4550.h*

Esta es la librería propia del microcontrolador en uso. En ella se definen todas las variables internas así como el direccionado de sus pines.

- *HDM64GS12.c*

Esta librería se utiliza para enviar datos y controlar un *LCD* gráfico específico, concretamente el modelo HDM64GS12. Cabe destacar que este no es el modelo de *LCD* utilizado en el presente proyecto, sino el JHD12864E, pero puesto que este último no dispone de librería propia, y ambos dispositivos utilizan el mismo controlador, el KS0108 de *Samsung*, el uso de esta librería es completamente compatible con el desarrollo del programa.

Entre otras cosas permite definir el modo de trabajo conocido como *FAST\_GLCD*, que consiste en trabajar sobre la memoria *RAM* creando en ella la imagen virtual a mostrar en el visualizador, y permitiendo un volcado de datos mayor al *LCD* actualizando como consecuencia la pantalla de una manera mucho más rápida. Se ha utilizado esta tecnología durante la programación del prototipo, obteniendo buenos resultados de refresco. Cabe mencionar que el uso de este modo de trabajo requiere que el microcontrolador disponga de 1 Kb de *RAM* exclusivo para este proceso. Este hecho influyó en la decisión de escoger el PIC18F4550 como microcontrolador final a utilizar.

- *GRAPHICS.c*

Esta librería ha resultado de mucha utilidad en el diseño, al permitir incluir funciones para dibujar líneas, rectángulos, cuadrados, círculos y texto en el *display*. Para ello se debe definir previamente una función un que dibuje un solo pixel, que es la siguiente:

glcd\_pixel (x, y, color)

donde *x* es la coordenada horizontal, e *y* es la coordenada vertical, y *color* es el bit que define si el pixel está pintado (1) o no (0). Se parte de que la posición (0,0) es la esquina superior izquierda de la pantalla y la (127, 63) es la esquina inferior derecha de la pantalla.

Sus funciones más son las siguientes:

- *glcd\_line*(*x1*, *y1*, *x2*, *y2*, *color*): dibuja una línea desde el punto (*x1*, *y1*) hasta el (*x2*, *y2*) con el color *ON* u *OFF*.
- *glcd\_rect* (*x1*, *y1*, *x2*, *y2*, *fill*, *color*): dibuja un rectángulo con una esquina en el punto (*x1*, *y1*) y la otra en (*x2*, *y2*), tanto el relleno (*fill*) como el color del borde pueden ser *ON* u *OFF*.
- *glcd\_bar*(*x1*, *y1*, *x2*, *y2*, *width*, *color*): dibuja línea de grosor variable desde el punto (*x1*, *y1*) hasta el (*x2*, *y2*) con el color *ON* u *OFF*, la variable *width* define el número de pixeles de anchura de la línea.



- `glcd_circle(x, y, radius, fill, color)`: dibuja un círculo con centro en (x, y), el radio en número de píxeles viene determinado por la variable *radius*, tanto el relleno (*fill*) como el color del borde pueden ser *ON* u *OFF*.
  - `glcd_text57(x, y, textptr, size, color)`: Escribe el texto contenido por la variable tipo *char textptr*, con la coordenada izquierda superior del primer carácter en el punto (x, y). Los caracteres tienen 5 píxeles de ancho y 7 de alto. La variable *size* define la escala del texto y *color* de las letras puede ser *ON* u *OFF*.
- *MATH.c*

Es una librería especializada en realizar diversos cálculos matemáticos complejos. Es necesaria su inclusión en nuestro diseño para permitir calcular los senos y cosenos que describen la trayectoria recorrida por la pelota.

- *STDLIB.h*

La principal particularidad de esta librería es que permite inicializar una semilla aleatoria. Este hecho es de suma importancia en el inicio de partida, puesto que define aleatoriamente las trayectorias a seguir por la pelota o pelotas al inicio del nivel. Cabe destacar que esta librería necesita a su vez de otras dos librerías más, *stdlib.h* y *string.h*, ambas relacionadas con el tratamiento de variables. Por último mencionar que la librería *string.h* requiere de la inclusión de otra librería, la *ctype.h*, también relacionada con el tratamiento de variables.

- *TONES.c*

Esta es la librería encargada de generar tonos en el PIC, se basa en el principio de que las notas musicales tienen una frecuencia determinada. Así que tras definir el *pin* que será utilizado como salida de audio, el usuario define en su programa la nota que quiere y su duración, y el PIC lo que hace es variar el valor de ese *pin* a la frecuencia correspondiente a la nota definida. Repitiendo este proceso se pueden llegar a generar melodías complejas.

#### 5.4.3. Configuración del PIC

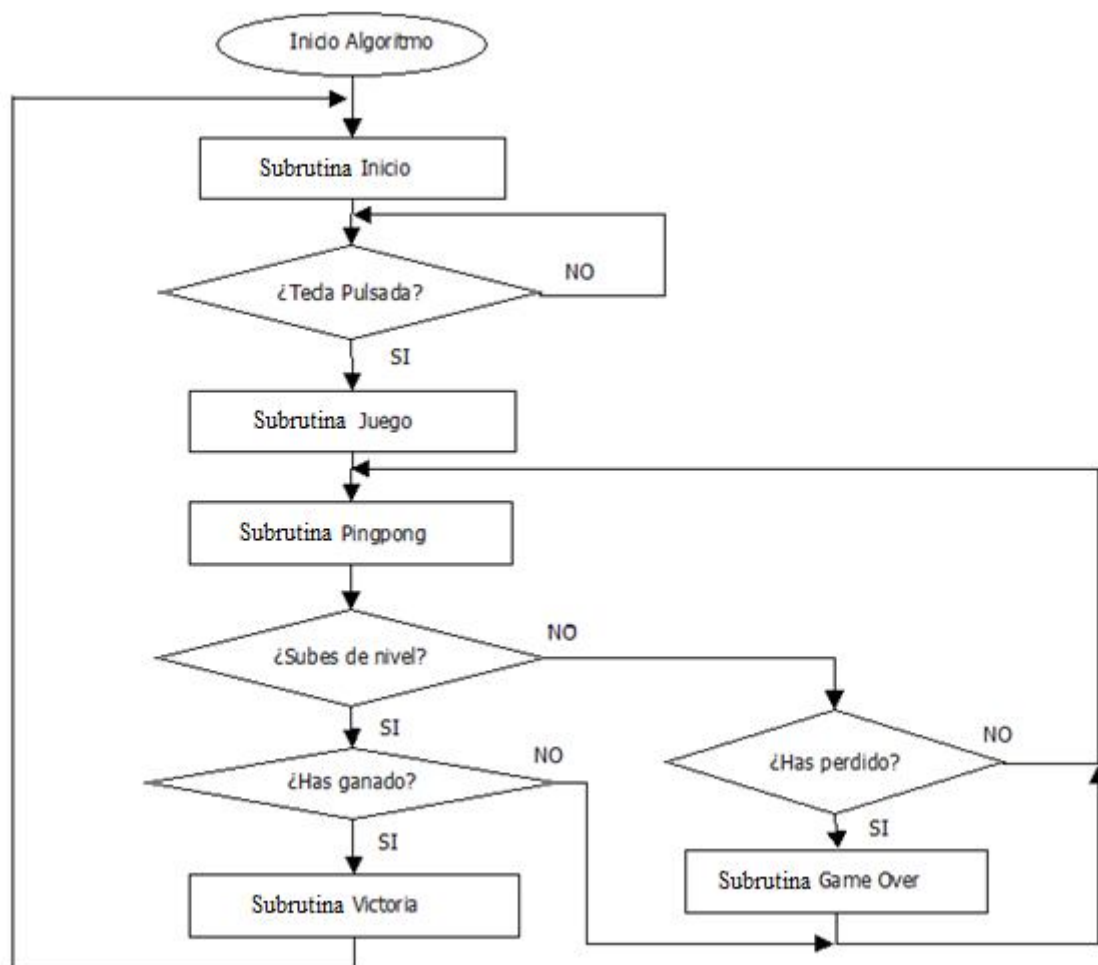
Para el correcto funcionamiento del PIC, se deben configurar previamente mediante software algunas directivas, fuses y puertos de entrada/salida.

En primer lugar hay que configurar los puertos según vayan a ser utilizados como entrada o como salida como se ha explicado en apartados anteriores. Este es también el momento de habilitar las interrupciones externas en el caso de que se utilicen. Al principio se deben configurar los fuses según las características del proyecto que se quiera realizar, por ejemplo utilizar o no

una señal de reloj externa. En este caso la señal de reloj es externa y es de 12 MHz.

#### 5.4.4. Diagrama de flujo Deadly Pong

En este apartado se presenta el diagrama de flujo del videojuego realizado que ayudará a entender su desarrollo representando su algoritmo de ejecución. Cabe destacar que todas las subrutinas forman parte de la misma rutina principal.



**Figura 70.** Diagrama de flujo del Deadly Pong (fuente: propia).

#### 5.4.5. Rutinas y Subrutinas de Deadly Pong

A continuación se procede a explicar la función de cada una de las rutinas y subrutinas utilizadas en el diseño del videojuego *Deadly Pong*. Se empezará por las subrutinas al estar relacionadas con el diagrama de flujo del apartado anterior.

- *Subrutina Inicio*

Se encarga de inicializar todas las variables requeridas para el comienzo de la partida. Inicializa además el *display GLCD* y muestra en pantalla la imagen de portada del videojuego mientras reproduce la melodía de victoria, también utilizada como melodía de inicio. Muestra además un falso indicador de carga en un guiño del proyectista a esos antiguos y no tan antiguos juegos de ordenador en los que los tiempos de carga eran excesivamente largos. Finalmente queda a la espera de que el usuario pulse un cualquier botón para ir a la siguiente subrutina.

- *Subrutina Juego*

Es una subrutina de transición que en principio está vacía, pero ha sido utilizada durante el desarrollo y simulación del juego para acceder directamente al nivel deseado por el programador y hacer diversas comprobaciones, y se ha conservado por la misma función. Puesto que durante el transcurso normal de la partida está vacía, el programa salta a la siguiente subrutina.

- *Subrutina pingpong:*

Es la subrutina donde se desarrolla el juego. En ella primeramente se inicializa la semilla aleatoria de las variables que controlan los ángulos de salida de la o las pelotas. Después se calculan los ángulos definitivos basados en el sistema aleatorio y se dan las posiciones iniciales de las pelotas. Una vez hechos los pasos anteriores, se entra en un bucle infinito que va sumando puntos por cada ciclo, y según los puntos que tengamos, iremos avanzando de nivel. Es en este bucle infinito donde se desarrolla la partida, la idea está en que tanto las pelotas como la pala están en movimiento, y por tanto a cada ciclo el microprocesador manda la señal de borrar el *GLCD* y de volver a imprimirlo, permitiendo ver la nueva posición de los objetos. Seguidamente se muestra un fragmento de código que hace referencia a este suceso.

```
glcd_circle((int)x,(int)y, 1, YES, Off);           // borramos pelota

x=(x+(3*cos(theta)));y=(y-(3*sin(theta))); //nuevas coordenadas

glcd_circle((int)x,(int)y, 1, YES, ON);           // pintamos pelota
```

También se tiene en cuenta el fenómeno del rebote de la pelota. En concreto cada vez que la o las pelotas llegan a los límites superior, izquierdo, y derecho, rebotan con un ángulo de salida complementario al que entraron. Obviamente el rebote también afecta a la pala, pero en este caso el ángulo de salida de la pelota variará según la zona de la pala en la que rebotara. Por ejemplo:

```
else if((y>58)&&(x>(m+4))&&(x<=(m+8))) //si la posición de la pelota llega al límite inferior
{
    //entre los 4 y 8 píxeles de la pala
    zumbador(); //se activa el sonido del rebote
    theta=-theta; //se hace el ángulo complementario
    theta=theta+0.2; //y además se desvía 0,2 radianes
}
```

Si se consigue aguantar lo suficiente sin perder, subiremos de nivel, y con ello se incrementará la dificultad. En concreto los aumentos de dificultad según el nivel son los siguientes:

- o Nivel 1: Una bola a velocidad 3.
- o Nivel 2: Dos bolas a velocidad 3.
- o Nivel 3: Dos bolas a velocidad 4.
- o Nivel 4: Tres bolas a velocidad 4.
- o Nivel 5: Tres bolas a velocidad 5.

Si se consigue llegar a la puntuación de victoria, iremos a la subrutina victoria. A continuación se incluyen las puntuaciones necesarias para cada nivel:

- o Nivel 1: 1 punto.
- o Nivel 2: 300 puntos.
- o Nivel 3: 600 puntos.
- o Nivel 4: 1000 puntos.
- o Nivel 5: 1300 puntos.
- o Victoria: 1400 puntos.

Si por el contrario la pelota llega al margen inferior y no se consigue hacerla rebotar con la pala, saltamos a la subrutina *gameover*.

Cada vez que la pelota rebota se llama a la rutina zumbador, encargada de realizar el efecto de sonido correspondiente.

Al inicio de cualquier nivel, se muestra una imagen de nivel mediante la rutina *image*.

- *Subrutina victoria:*

Esta subrutina solo se da en el caso de que se supere el nivel 5. En ella se reproduce la melodía de victoria y se felicita al jugador mediante imágenes. Finalmente se muestran los créditos y se vuelve a la subrutina inicio.

- *Subrutina gameover:*

Esta subrutina se ejecuta cada vez que la bola llega al límite inferior del área de juego. Se reproduce la melodía de perder bola y se muestra en pantalla el número de vidas que le quedan al jugador. Si el jugador aun tiene vidas lo manda de vuelta a la subrutina pingpong para que continúe jugando en el nivel en el que se encontraba, si por el contrario ya no dispone de vidas, se carga la imagen de derrota y se muestra la puntuación realizada junto con el récord de esa sesión. Finalmente se muestra la imagen de créditos y se vuelve a la subrutina pingpong.

Ahora se detallarán las rutinas que incluye el programa, que son las siguientes:

- *Rutina main (programa principal)*

Esta rutina está formada íntegramente por todas las subrutinas descritas anteriormente.

- *Rutina image*

La principal tarea de esta rutina cargar las imágenes de la memoria ROM a la RAM para poder mostrarlas por pantalla. La selección de imagen se hace mediante una instrucción *CASE*. Las imágenes están guardadas en el fichero *imagenes.c*, y hay un total de diez imágenes de 1024 bytes cada una.

- *Rutina tono\_victoria*

Esta rutina se encarga de reproducir el tono de victoria guardado en ella.

- *Rutina tono\_derrota*

Igual que la rutina anterior, su única finalidad es reproducir el tono de derrota que almacena.

- *Rutina tono\_perder\_bola*

Por último esta es la tercera rutina encargada de generar un tono musical, y que se acciona cada vez que el jugador pierde una vida.

- *Rutina zumbador*

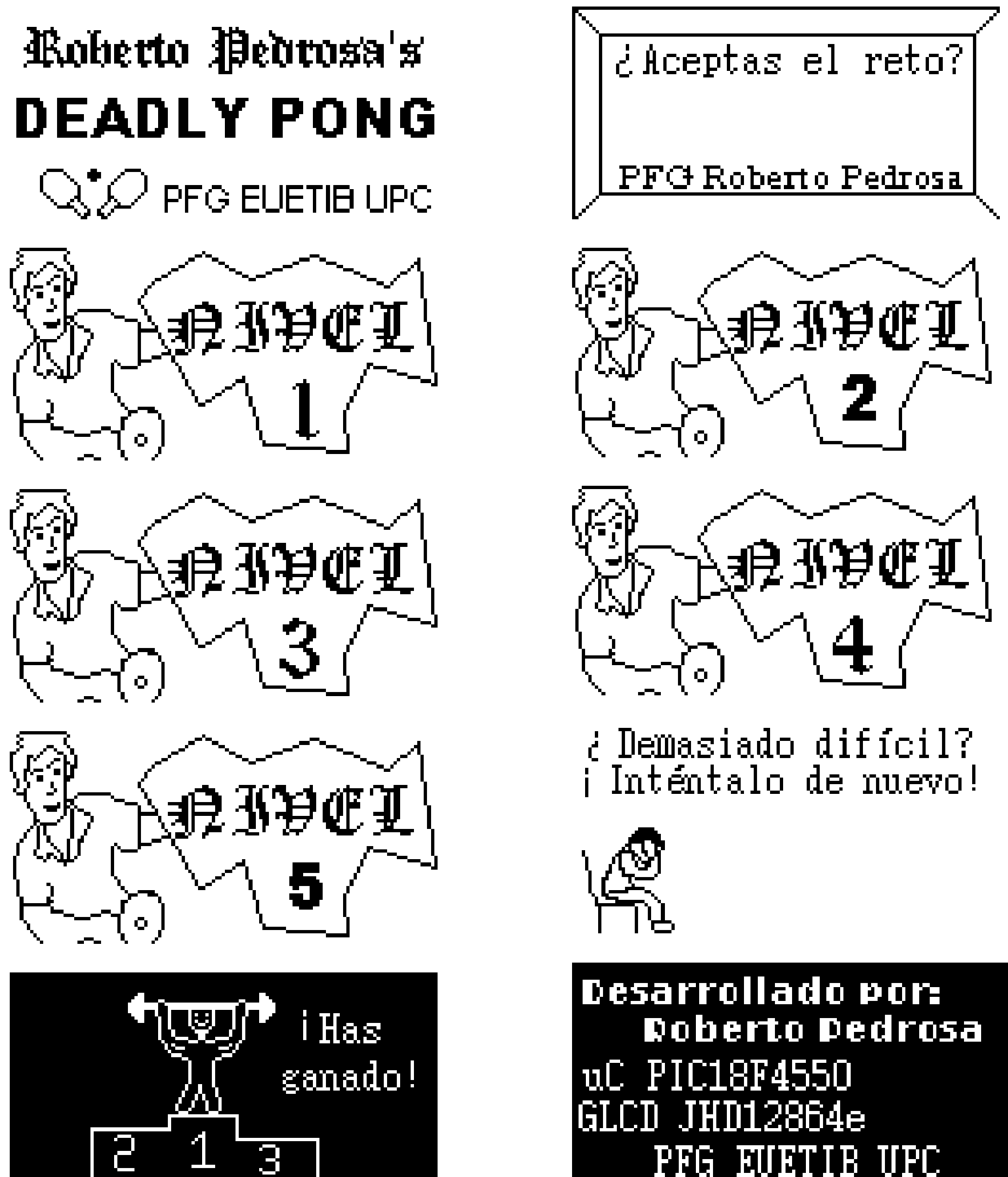
Esta rutina simplemente acciona el zumbador durante 5 ms para hacer el efecto de sonido del rebote de la bola.

#### 5.4.6. *Imágenes del Deadly Pong*

Este apartado recoge las imágenes que aparecen durante la partida. Estas imágenes han sido creadas por el proyectista mediante el *software Bitmap2LCD*. Esta herramienta de diseño te permite dibujar sobre una rejilla del tamaño de la pantalla *LCD* que se vaya a utilizar, y cuando se

tiene la imagen deseada, el programa genera el código en C correspondiente a esa imagen, para poder ser cargado directamente al microprocesador y de él al *LCD*.

Las imágenes que se han creado son las siguientes:



**Figura 71.** Conjunto de imágenes desarrolladas para el proyecto (fuente: propia).

#### 5.4.7. Melodías del Deadly Pong

Durante la partida se pueden escuchar tres melodías diferentes creadas por el proyectista:

- *Tono de victoria*

Este tono está inspirado en uno de los clásicos de los videojuegos, el tono de victoria de la saga *Final Fantasy*. Puesto que se inspira pero no es igual, el uso de esta melodía no infringe ningún derecho sobre la propiedad intelectual.

Este tono está compuesto por las siguientes notas con su respectiva duración:

Re' (250ms), Re'(250ms), Re'(250ms), Re'(500ms), Sib (500ms), Do'(250ms), Re'(250ms), Do'(250ms), Re'(500ms).

- *Tono de derrota*

Este tono no tiene ninguna inspiración particular. Simplemente se ha intentado crear una melodía empática que tenga que ver con el hecho de perder.

El tono está compuesto por las siguientes notas con su respectiva duración:

Do'(250ms), La(250ms), Fa(250ms), Re(250ms), Do(1000ms).

- *Tono de perdida de vida*

Por último se ha decidido incluir a parte del tono de derrota, otro tono que suene cada vez que el jugador pierde una vida. Es un tono satírico que no pretende otra cosa que enfatizar y motivar al jugador.

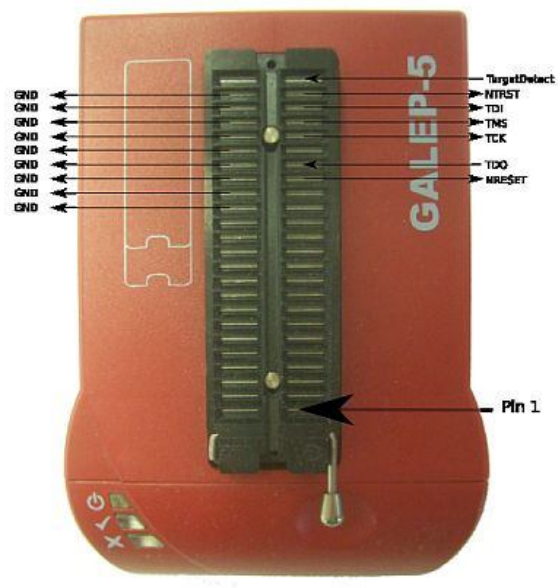
El tono está compuesto por las siguientes notas con su respectiva duración:

Mi(500ms), Do(500ms).

### 5.5. Entorno de grabación del PIC

El volcado del programa al microcontrolador PIC se ha hecho mediante el grabador *Galep 5*. Es un tipo de grabador que permite interactuar con el PC a través del puerto USB.

Mediante su entorno de grabación, cargamos la el archivo .hex resultante de la compilación en el microcontrolador PIC18F4550. El *software* es muy intuitivo y fácil de utilizar.



**Figura 72.** Grabador GALEP-5 (fuente: <http://www.conitec.net/manual/galepx/en/g5ocd.htm>).



# **CAPÍTULO 6:**

## **CRONOGRAMA DE EJECUCIÓN**

Para la optimización de tiempo durante la ejecución del proyecto, se ha dividido en tareas independientes para realizar una estimación del tiempo necesario y una planificación eficiente.

### **6.1. Programa de ejecución**

La realización de este proyecto se puede desglosar en el siguiente conjunto de tareas:

- Búsqueda de información
- Estudio previo
- Selección de componentes
- Elaboración de un esquema electrónico inicial
- Análisis de las herramientas de software
- Diseño del videojuego
- Simulación del videojuego
- Corrección de errores
- Elaboración del esquema electrónico definitivo
- Adquisición de los componentes
- Testeo de los componentes
- Diseño de la placa de circuito impreso
- Montaje
- Grabación del PIC
- Comprobación de funcionamiento
- Posible corrección de errores final.
- Realización de posibles mejoras
- Elaboración de la memoria de proyecto
- Preparación de la defensa

## 6.2. Estimación de tiempo

Para la correcta planificación de tiempo de este proyecto se han tenido en cuenta las siguientes fechas:

- Fecha de inicio del proyecto: 13 de febrero de 2012.
- Fecha de entrega de la documentación del proyecto: 13 de junio de 2012.
- Fecha de inicio de las presentaciones: 29 de junio de 2012.

Para la correcta realización de este proyecto se le presta una dedicación completa, con lo cual el número total de días disponibles para su desarrollo es:

- 122 días desde la fecha de inicio hasta la fecha de entrega de la documentación del proyecto
- 16 días para preparar la defensa del proyecto desde la entrega de la documentación hasta el inicio de las presentaciones

Considerando los domingos como único día de descanso, el número de días totales dedicados al proyecto es de 118, y teniendo en cuenta una dedicación promedia de 5 horas y media al día, el número de horas totales empleado es de 649 horas.

Teniendo en cuenta que el proyecto final de grado es contabilizado como una asignatura de 24 créditos ECTS, y sabiendo que un crédito ECTS equivale a 25 horas de dedicación, se puede establecer que la dedicación teórica total tendría que ser aproximadamente 600 horas.

Contrastando estos dos resultados se observa que ambos valores son bastante parecidos, dando a entender que el alcance del proyecto ha sido el adecuado para este trabajo.

El cronograma se ha realizado con el *software Gantt Project*, que se distribuye gratuitamente y no requiere ningún tipo de licencia para su uso.

El cronograma completo se encuentra en los anexos.

# CAPÍTULO 7:

## PRESUPUESTO

Como su nombre indica, este capítulo recoge los gastos totales originados durante el desarrollo del proyecto.

### 7.1. Costes de los materiales

Para poder tener una previsión económica precisa y fiable se deben tener en cuenta todos los gastos directos e indirectos que ha generado la realización de este proyecto. Por ello se ha hecho la siguiente lista donde aparecen todos los costes del prototipo.

**Taula 2.** Lista de componentes y materiales utilizados en el proyecto y su coste total.

<i>Ctd.</i>	<i>Ref.</i>	<i>Descripción</i>	<i>Precio unitario (€)</i>	<i>Precio total (€)</i>
1	Microcontrolador	Microcontrolador PIC18F4550 de la marca <i>Microchip</i>	5,90	5,90
1	GLCD	Display JHD12864E, LCD gráfico monocromático de 128x64 pixeles	17,95	17,95
1	Altavoz	Microaltavoz de 0,3W y 8 ohms de <i>FoneStar</i>	6,36	6,36
1	PCB	Placa emulsionada positiva de fibra de vidrio 100x160	4,59	4,59
1	Zumbador	Zumbador piezocerámico de 3KHz alim. 3-24V y 10mA	1,19	1,19
3	Botones	Pulsador Verde 125V 3A	2,19	6,57
1	Interruptor	Interruptor Bipolar 6A/250VAC	1,12	1,12

1	Caja	Caja de plástico POLIBOX 220x135x75mm	21,83	21,83
1	Alimentación	Fuente de alimentación externa estabilizada continua de 5V	15,95	15,95
5	R47	Resistencia de carbono prensado de valor 47 ohmios, tolerancia $\pm 5\%$ y 0,25W de potencia máxima disipada.	0,02	0,10
5	R100	Resistencia de carbono prensado de valor 100 ohmios, tolerancia $\pm 5\%$ y 0,25W de potencia máxima disipada.	0,02	0,10
5	R140	Resistencia de carbono prensado de valor 140 ohmios, tolerancia $\pm 5\%$ y 0,25W de potencia máxima disipada.	0,02	0,10
5	R10k	Resistencia de carbono prensado de valor 10000 ohmios, tolerancia $\pm 5\%$ y 0,25W de potencia máxima disipada.	0,02	0,10
1	Potenciómetro	Potenciómetro ajustable de 10 k $\Omega$ .	0,49	0,49
5	C18pF	Condensador cerámico de 18 pF, y tensión máxima 63V.	0,02	0,10
5	C100nF	Condensador cerámico de 100 nF, y tensión máxima 63V.	0,02	0,10
1	Buffer	Circuito integrado 74HC4049 buffer HEX inversor HS-CMOS.	0,70	0,70
1	Z16	Zócalo para circuito integrado de 16 pines.	0,10	0,10
1	Z40	Zócalo para circuito integrado de 40 pines.	0,23	0,23
1	Transistor	Transistor BJT tipo NPN modelo BD135 45V 1,5 TO126.	0,08	0,08
1	Tornillos	Blíster de 20 tornillos de métrica M3 de 20mm de acero.	1,50	1,50
1	Bus20	Cable plano de bus de 20 conductores (1 metro)	1,20	1,20
1	Con20M	Conector macho para cable plano de bus de datos con 20 conexiones.	0,75	0,75
1	Con20H	Conector hembra para cable plano de bus de datos con 20 conexiones.	0,75	0,75

1	Con2M	Conector macho de alimentación con dos conexiones compatible con la fuente de alimentación	0,60	0,60
1	Cristal	Cristal de cuarzo de 12MHz de resonancia	0,34	0,34
1	LED verde	Diodo LED de color verde 2,2V	0,10	0,10
1	Brocas	Juego de brocas de entre 0,5 y 1,5 mm para perforar la PCB	4,95	4,95
1	Varios	Estaño, pasta de soldadura, etc.	5,00	5,00
			Subtotal	98,85
			I.V.A. (18%)	17,79
			<b>TOTAL</b>	<b>116,64</b>

## 7.2. Costes de la ingeniería

Los costes de la ingeniería consisten en el coste de los sueldos del personal que ha realizado la ejecución del proyecto. El coste total se desglosa según las horas dedicadas a cada tarea específica, partiendo del mismo sueldo base.

Se parte de la base que el sueldo medio en Cataluña de un ingeniero técnico industrial especializado en Electrónica Industrial recién titulado, es de 15€ la hora, según datos del CETIB (Colegio de Ingenieros Técnicos Industriales) respecto al año 2011.

**Taula 3.** *Lista de tareas y su respectivo coste total.*

<i>Actividad</i>	<i>Horas dedicadas (horas)</i>	<i>Coste total (€)</i>
Búsqueda de información	30	450
Estudio previo	50	750
Selección de componentes	20	300
Elaboración de un esquema electrónico inicial	10	150
Análisis de las herramientas de software	20	300
Diseño del videojuego	150	2250

Simulación del videojuego	20	300
Corrección de errores	30	450
Elaboración del esquema electrónico definitivo	10	150
Testeo de los componentes	20	300
Diseño de la placa de circuito impreso	10	150
Montaje	30	450
Grabación del PIC	4	60
Comprobación de funcionamiento	10	150
Corrección de errores final	30	450
Realización de posibles mejoras	50	750
Subtotal	494	7410
	IVA (18%)	1334
	<b>TOTAL</b>	<b>8744</b>

### 7.3. Otros costes

En este apartado se engloban los costes proporcionales del uso de diverso equipo electrónico e informático así como de las licencias del software utilizado.

Ya que el los dispositivos electrónicos se pueden amortizar con más proyectos a parte de este, se pone una cantidad proporcional a su uso para determinar su coste final.

**Taula 4.** Lista de costes proporcionales del uso de software coste total.

<i>Ctd.</i>	<i>Ref.</i>	<i>Descripción</i>	<i>Precio unitario (€)</i>	<i>Precio total (€)</i>
0,05	PC	Uso de entorno informático.	500	25
0,10	Grabador	Grabador GALEP-5.	80	8
0,05	OrCAD	Se utiliza con licencia de demostración, así que es gratuito.	0	0
0,05	Proteus	Para el uso de <i>Proteus</i> Nivel 1 si que se requiere licencia.	481	24
0,05	CCS C	Este software también requiere de la adquisición de licencia para poder ser utilizado.	399	20
0,5	Bitmap2LCD	El uso de este software requiere de una licencia de uso que se puede adquirir en la página web del producto.	65	33
1	Insolación	Servicio externo de insolación de placas de circuito impreso, como se ha hecho en la facultad es gratuito	0	0
			Subtotal	110
			IVA (18%)	20
			<b>TOTAL</b>	<b>130</b>

## 7.4. Costes totales

El coste final del proyecto será de:

***Taula 4. Lista de costes totales.***

<i>Concepto</i>	<i>Coste total (€)</i>
Costes de los materiales	116,64
Costes de la ingeniería	8744,00
Otros costes	130,00
<b>TOTAL</b>	<b>8990,64</b>

Así pues el coste total de la realización del proyecto será de 8990,64 €.



# CAPÍTULO 8:

## POSIBLES MEJORAS Y CONCLUSIONES

En este apartado se estudiarán posibles futuras mejoras a realizar en el diseño, y se hará una pequeña conclusión sobre el mismo.

### 8.1. Posibles mejoras

El diseño tiene un gran margen de mejora, tanto *hardware* como *software*. Se procede a enumerar las posibles mejoras que el proyectista tiene pensado realizar en el futuro.

#### 8.1.1. Mejoras en hardware

En primer lugar, si se desea desarrollar un juego más potente o complejo, algunos de los módulos utilizados deberán ser sustituidos, por ejemplo se podría sustituir el *GLCD* gráfico monocromático por uno que permitiese mostrar gráficos a color. Se podría añadir un módulo de memoria externo que permitiera almacenar más imágenes e instrucciones y poder aumentar la complejidad del juego realizado.

Otra posibilidad es añadir la opción de que la consola sea del tipo multijugador, debiendo añadir por tanto más botones para que puedan jugar 2 o más jugadores. Incluso se podrían añadir más botones en el modo de un jugador para diseñar juegos que requieran más complejidad de control, por ejemplo añadiendo un botón de disparo, o dos de desplazamiento hacia arriba y hacia abajo, las posibilidades son infinitas.

Respecto al audio, la limitación del sistema actual es que no puede reproducir melodías mientras se juega debido a que interrumpe a las instrucciones del juego para reproducir el audio. Una posible solución es incorporar otro microcontrolador que se encargue únicamente del audio, permitiendo al microcontrolador inicial controlar el desarrollo de la partida

mientras el segundo se encarga íntegramente del audio, permitiendo hacer reproducir así melodías durante la partida ininterrumpidamente.

### *8.1.2. Mejoras en el software*

Las mejoras en el *software* que se pueden realizar son obvias, se pueden diseñar juegos más potentes, o hacer que el propio sistema sea capaz de reproducir diferentes juegos que tenga en la memoria interna, creando un menú de selección de juego inicial. También se podría hacer que el jugador que haga un récord pueda inscribir su nombre junto a su puntuación en el bloque de memoria EEPROM.

## 8.2. Conclusión personal

La realización de este proyecto final me ha servido para consolidar todos los conocimientos adquiridos durante la carrera de una manera muy interesante y divertida, y adquirir muchos conocimientos nuevos útiles en mi futura vida laboral. Me considero un apasionado de los videojuegos y por eso cuando vi este proyecto ofertado no dudé en elegirlo para poder diseñar mi propia videoconsola, algo que me ha interesado desde pequeño.

La dificultad se hizo patente desde el principio pero cuando el tema es afín al autor la dificultad pasa a un segundo plano y permite al alumno concentrarse y salvar las dificultades encontradas mientras aprende y se divierte.

Otro aspecto muy positivo de este proyecto, es que me ha brindado la oportunidad de trabajar con un tipo de microcontrolador que no hemos visto durante nuestro plan de estudios, el PIC, y que es hoy en día de los más usados en el ámbito de la electrónica y de la informática industrial.

Realizar este trabajo ha sido muy positivo para mí porque me ha permitido relacionar un posible futuro laboral con una de mis formas de ocio preferidas, despertándome las ganas de probar suerte en este sector en el futuro, y sobretodo a seguir formándome para poder desarrollar en el futuro proyectos similares de mayor complejidad.

Finalmente cerrar mi paso por la facultad con un proyecto final tan ameno ha sido la guinda perfecta para acabar mis estudios de Grado en Ingeniería Electrónica Industrial y Automática.

# **CAPÍTULO 9: BIBLIOGRAFÍA**

## **9.1. Referencias bibliográficas**

Microchip. Microchip Technology Inc. [En línea]. [Consulta: 10 abril 2012]. Disponible en: <<http://www.microchip.com/>>.

Facultat d'Informàtica de Barcelona - UPC. Retroinformàtica. Història dels videojocs. [En línea]. [Consulta 16 marzo 2012]. Disponible en: <<http://www.fib.upc.edu/retroinformatica/historia/videojocs.html>>.

Microcontroladores PIC. Cursos, tutoriales, programas, datasheet, esquemas y proyectos con microcontroladores PIC. [En línea]. [Consulta 16 marzo 2012]. Disponible en: <<http://www.microcontroladorespic.com/>>.

Foros de electrónica: Comunidad de Electrónicos. Control de Display LCD con microcontrolador PIC. [En línea]. [Consulta 20 marzo 2012]. Disponible en: <<http://www.forosdeelectronica.com/f24/control-display-lcd-microcontrolador-pic-201/>>.

Foros de electrónica: Comunidad de Electrónicos. Como controlar un LCD grafico. [En línea]. [Consulta 20 marzo 2012]. Disponible en: <<http://www.forosdeelectronica.com/f24/controlar-lcd-grafico-23982/>>.

## 9.2. Bibliografía de consulta

### 9.2.1. Artículos de revistas

Pocket Pong. Un juego antiguo con hardware moderno. Elektor, N° 291, págs. 58-61, 2004. ISSN 0211-397X.

### 9.2.2. Libros

Marco A. Peña Basurto, JM Cela. *Introducción a la programación en C*. 1ª ed. Barcelona: Ediciones UPC, 2000.

Saltzman, Marc. *Cómo diseñar videojuegos*. [trad.] Verónica Canales y Lorenzo F. Díaz. 3ª ed. Barcelona: Norma, 2001.

Lehmann, Stefan y Harth, Wolfram. *Microcontroladores PIC: prácticas de programación*. [trad.] Virginia Pérez Moreno. 2ª ed. Barcelona: Marcombo, 2008.

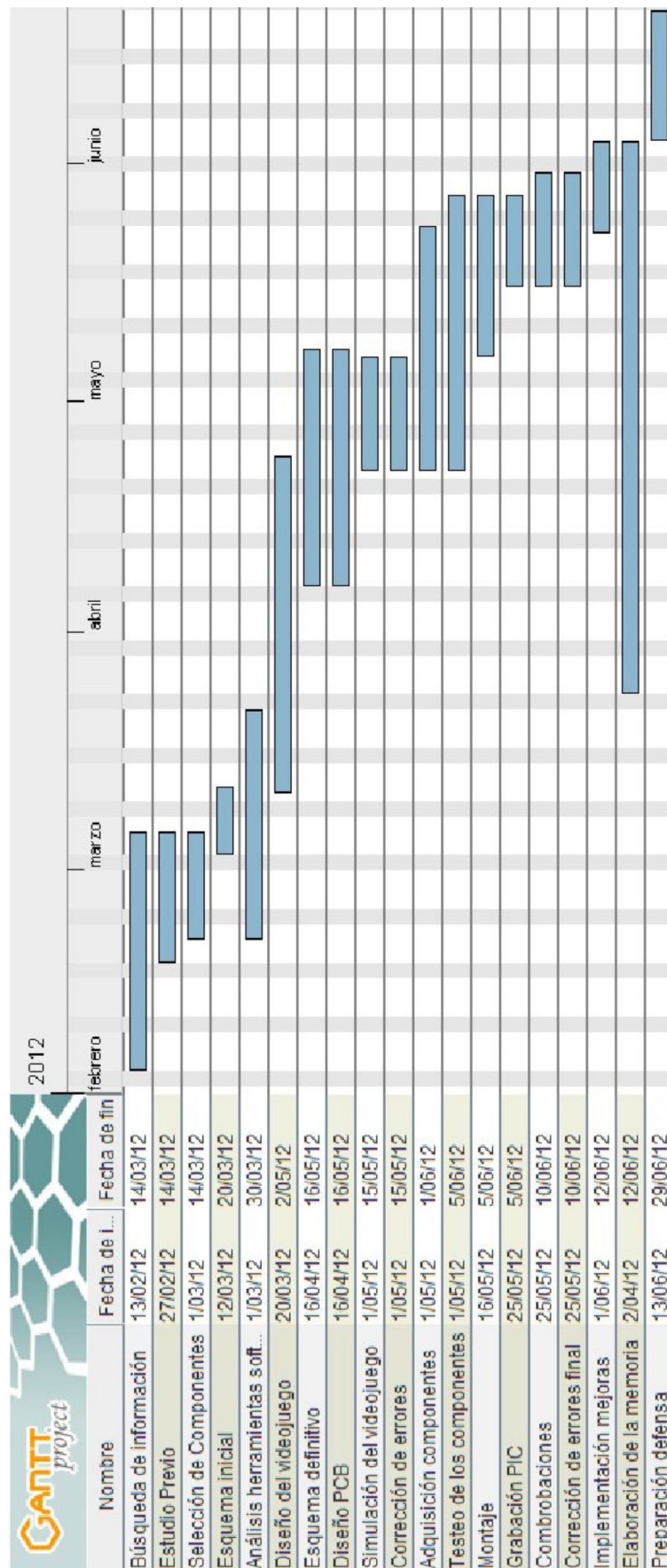
Kent, Steven L. *The First quarter: a 25-year history of videogames*. Bothell, WA: BWD Press, 2000.

García Breijo, Eduardo. *Compilador C CCS y simulador Proteus para microcontroladores PIC*. 1ª ed. Madrid: Marcombo, 2006.

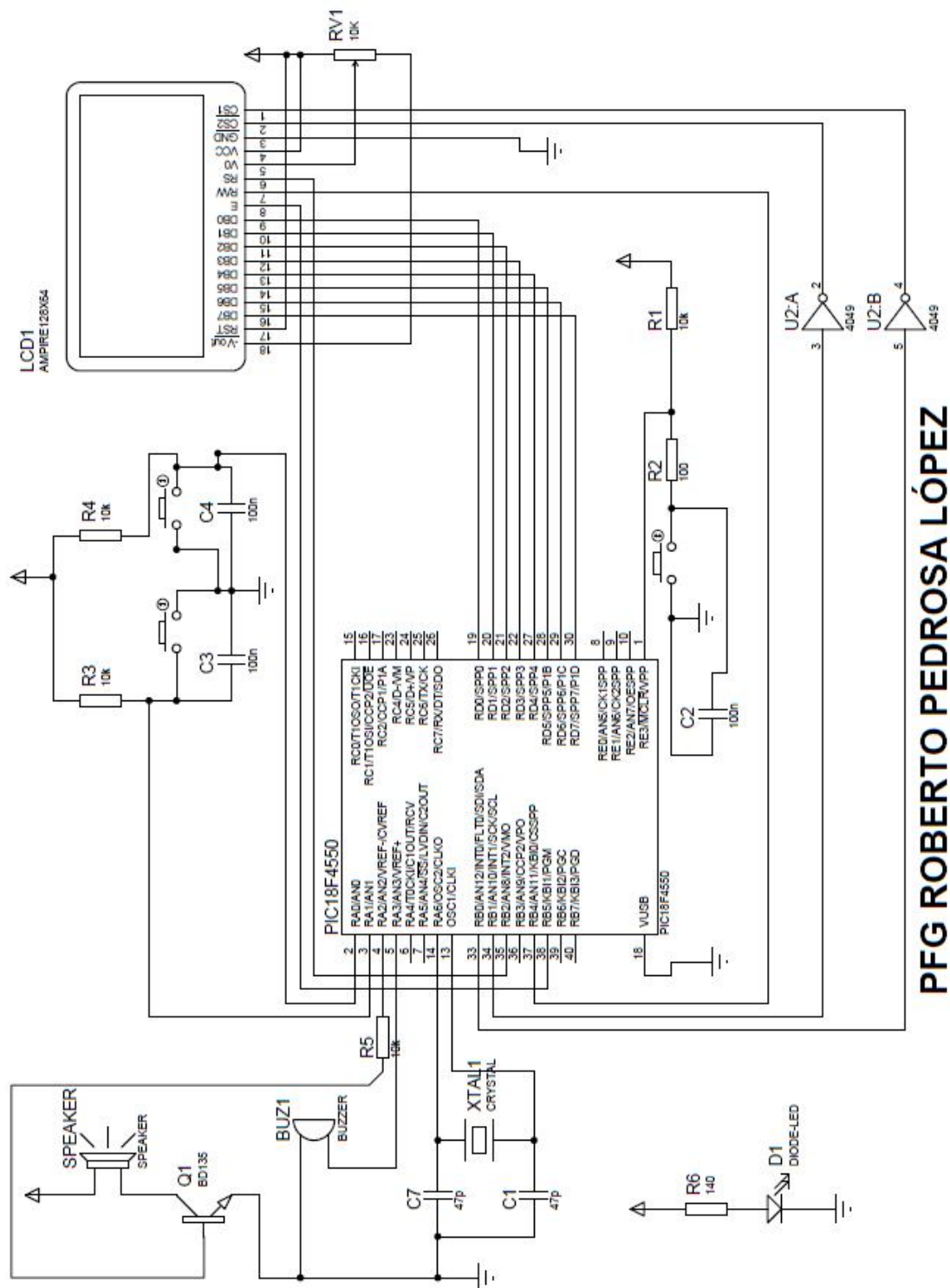
Valdés, Fernando y Pallas, Ramón. *Microcontroladores: Fundamentos y aplicaciones con PIC*. 1ª ed. Madrid: Marcombo, 2007.

# **ANEXO A: ESQUEMAS Y PLANOS**

- Diagrama de Gantt del proyecto



Esquema eléctrico del conjunto:



PFG ROBERTO PEDROSA LÓPEZ

Esquema PCB:

PFG ROBERTO PEDROSA LOPEZ EUETIB

